REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE June 30, 2007	3. REPORT TYPE AND DATES COVERED Final Report June 1, 2001 to May 31, 2007

4. TITLE AND SUBTITLE

Mathematics of Failures in Complex Systems: Characterization and Mitigation of Service Failures in Complex Dynamic Systems

5. FUNDING NUMBERS

DAAD19-01-1-0646

6. AUTHOR(S)

Dr. Asok Ray, Principal Investigator

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

College of Engineering
Pennsylvania State University, University Park, PA 16802

8. PERFORMING ORGANIZATION REPORT NUMBER

FR CSF07001

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

42245-MA-MUR

11. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Objectives:
The goal of this MURI project is to enhance dependability of complex engineering systems that include both human-operated and unmanned machinery. It focuses on pervasive fault tolerance based on the emerging information. The specific objectives are:
1. Early detection of small anomalies in nonlinear and nonstationary dynamical systems
2. Analysis and synthesis of robust decision and control systems to restrict the system behavior within desired specifications
3. Development of high-fidelity test apparatuses and experimental validation of theoretical results

Results:
Principles of Automata Theory and Statistical Signal Processing are applied to retrieve relevant information for failure prognosis and decision & control of complex dynamical systems. Experimental research has been conducted to validate the theoretical results at the following three laboratories:
• Electromechanical Systems
• Networked Robotic Systems
• Engineering Systems Simulation

14. SUBJECT TERMS

Complexity analysis, nonlinear dynamics, chaos, fractional-dimensional dynamics, small change detection, formal languages, supervisory control, life extension of rotorcraft, fault tolerance in robotic networks, computer system failures, and risk analysis, computer user authentication

15. NUMBER OF PAGES

142

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Enclosure 1

MASTER COPY: PLEASE KEEP THIS "MEMORANDUM OF TRANSMITTAL" BLANK FOR REPRODUCTION PURPOSES. WHEN REPORTS ARE GENERATED UNDER THE ARO SPONSORSHIP, FORWARD A COMPLETED COPY OF THIS FORM WITH EACH REPORT SHIPMENT TO THE ARO. THIS WILL ASSURE PROPER IDENTIFICATION. NOT TO BE USED FOR INTERIM PROGRESS REPORTS; SEE PAGE 2 FOR INTERIM PROGRESS REPORT INSTRUCTIONS.

MEMORANDUM OF TRANSMITTAL

U.S. Army Research Office
ATTN: AMSRL-RO-BI (TR)
P.O. Box 12211
Research Triangle Park, NC 27709-2211

□ Reprint (Orig + 2 copies)

□ Manuscript (1 copy)

□ Technical Report (Orig + 2 copies)

☑ Final Progress Report (Orig + 2 copies)

□ Related Materials, Abstracts, Theses (1 copy)

CONTRACT/GRANT NUMBER: C-DAAD19-01-1-0646

REPORT TITLE: Characterization and Mitigation of Service Failures in Complex Dynamic Systems

is forwarded for your information.

SUBMITTED FOR PUBLICATION TO (applicable only if report is manuscript):

Sincerely,

Dr. Asok Ray, Principal Investigator

PROJECT:	MURI on Mathematics of Failures in Complex Systems
REPORT TITLE:	**CSF Final Report**
PRINCIPAL INVESTIGATOR:	*Dr. Asok Ray*
OTHER KEY CONTRIBUTORS:	*Dr. Shashi Phoha* *Dr. Joseph Horn* *Dr. Eric Keller* *Dr. V.V. Phoha* *Dr. D. Siewiorek* *Dr. K. Trivedi*
REPORT NUMBER:	TR CSF07001
STATUS:	Final Report for the Years 2001-2007
DATE:	June 15, 2007

FOREWORD

The MURI project titled *Characterization and Mitigation of Failures in Complex Dynamical Systems* was started on June 1, 2001 and ended on May 31, 2007. It was conducted under the leadership of Pennsylvania State University, University Park, PA and other participating universities were Carnegie-Mellon, Louisiana Tech, and Duke. This research project has developed a novel approach to anomaly characterization and real-time decision-making in complex dynamical systems to achieve *pervasive fault tolerance* based on the emerging information. Extensive experimental research has been conducted to validate the theoretical results. Special-purpose laboratory apparatuses have been procured. On the experimental side of the CSF MURI project, three laboratories have been constructed at Penn State under equipment support from DURIP grants for five consecutive years.

The research work, conducted under this research, has resulted in publication of more than 125 scholarly publications, one research monograph, and completion of 15 doctoral dissertations. This MURI project has effectively transferred the newly developed technology to defense industry through five SBIR and STTR projects in collaboration with industry.

TABLE OF CONTENTS

LIST OF APPENDICES

STATEMENT OF THE PROBLEM STUDIED

The research, conducted under this MURI project, focused on characterization and real-time mitigation of *pervasive faults* in human-engineered complex dynamical systems. These systems, although architecturally similar to physical systems, are often structurally quite different. At the lowest level of decomposition, the macroscopic effects could be triggered by single fault manifestations of emerging physical defects in hardware, erroneous states of software, or human operator errors. Analytical methods have been developed and experimentally validated for sustaining order and normalcy by deriving and finding critical values of physical parameters, where the subsequent behavior of the macroscopic system could change abruptly.

For theoretical analyses, complex dynamical systems have been modeled as hybrid interacting automata, whose continuously varying dynamics characterize the physical process at the lowest level of abstraction. Discrete event models at the higher levels capture the cognitive response of the system to observed emerging phenomena. This concept has been used to utilize the dynamical behavior of structural materials in formulating damage mitigating control algorithms at the system level to enhance the life of critical mechanical components. The research has formulated analytical models of component interactions triggered by different types of failures to: (*i*) predict emerging pathological system behavior from time-series observations of events and their dynamic interactions, and (*ii*) formulate adaptive mechanisms to circumvent or mitigate the effects of pathological behavior. The research efforts have resulted in comprehensive characterization of pathological behavior, both syntactic and operational, through analysis of spatial-temporal patterns in databases of event/action dynamics as well as by extensive experimentation.

The research has provided a scientific basis for engineering dependability in military operations with a fundamentally new approach to engineering and operation of complex informational systems for *pervasive fault tolerance*. Instead of specifying parameters for worst-case design of system components, these systems are designed by specifying a scalable set of resources (components) that interact to support evolving operational needs of defense applications in a dynamic and uncertain environment.

SUMMARY OF THE MOST IMPORTANT RESULTS

Multidisciplinary University Research Initiative (MURI) Project

Characterization and Mitigation of Failures in Complex Dynamical Systems

Principal Investigator: Professor Asok Ray, Pennsylvania State University

Participating Universities: Carnegie-Mellon; Louisiana Tech; and Duke

The objective of the Complex Systems Failure (CSF) Multidisciplinary University Research Initiative **(**MURI) project has been to formulate and disseminate a knowledge base of science and technology for enhancing dependability of complex engineering systems that include both human-operated and unmanned machinery. It has developed a fundamentally new approach to anomaly characterization and real-time decision-making in complex dynamical systems to achieve *pervasive fault tolerance* based on the emerging information. Dependability of complex systems is achieved by identifying and mitigating the origins of chaos and disorder at a very early stage through dynamic coordination and control of the critical subsystems.

On the theoretical side of the CSF MURI project, complex dynamical systems (e.g., chaotic motion and bifurcation) are modeled from the perspectives of (Classical and Quantum) Statistical Mechanics and Automata Theory to retrieve relevant information for decision and control. In this approach, dynamical systems are represented by hybrid interacting automata where continuously varying information can be captured at various levels of abstraction. For example, this concept utilizes the information on multi-fractal behavior of ductile alloys in formulating damage mitigating decision & control algorithms at the system level to enhance the service life of critical electromechanical components in complex systems. The goal is to formulate analytical models of component interaction dynamics triggered by various types of exogenous disturbances and intrinsic failures to (*i*) predict emerging pathological behavior of the dynamical system from observations of time-series data and time-dependent discrete events; and (*ii*) formulate adaptive mechanisms to circumvent or mitigate the effects of pervasive faults.

Extensive experimental research has been conducted to validate the above theoretical results. Special-purpose laboratory apparatuses have been procured. On the experimental side of the CSF MURI project, the following three laboratories have been constructed at Penn State under equipment support from DURIP grants for five consecutive years:

- Electromechanical and Electronic Systems Laboratory
- Networked Robotic Systems and Signal Intelligence Laboratory
- Engineering Systems Simulation Laboratory

The research work, conducted under this MURI from June 1, 2001 to May 31, 2007, has been reported in archive journals, refereed conferences, patent disclosures, and research monographs, as listed below.

- Over fifty archive journal publications
- Seventy five refereed conference publications
- Three patent disclosures
- One research monograph and several book chapters

This MURI project has supported training of five postdoctoral scholars and has produced fifteen doctoral dissertations. A majority of the graduate students, upon completion of their doctoral education, have accepted research employment in DoD laboratories and defense contractors.

This MURI project has effectively transferred the newly developed technology to defense industry through five SBIR and STTR projects in collaboration with industry.

LIST OF PUBLICATIONS UNDER ARO SPONSORSHIP

(a) Papers published in peer-reviewed journals

1. A. Ray and S. Phoha, "Detection of Potential Faults via Multi-level Hypotheses Testing," ***Signal Processing***, Vol. 82, No. 6, June 2002, pp. 853-859.
2. D. Y. Chen and K. S. Trivedi, "Closed-Form Analytical Results for Condition-Based Maintenance," ***Reliability Engineering and System Safety***, Vol. 76, 2002, pp. 43-51.
3. D. Y. Chen, Y. Hong, and K. S. Trivedi, "Second Order Stochastic Fluid Models with Fluid Dependent Flow Rates," ***Performance Evaluation***, Vol. 49, 2002, pp. 341-358.
4. U. Tuntoolavest and J. J. Metzner, "Vector Symbol Decoding with Erasures, Errors and Symbol List Decisions," ***Integrated Computer-Aided Engineering***, Vol. 9, No. 2, 2002, pp. 101-116.
5. C. Delfino, A. Ray, K. N. Ivanov and F-B. Cheung, "A New Concept for Early Detection of BWR Instabilities," ***Trans. of American Nuclear Soc*. (*TransAO*)**, Vol. 88, June 2003, pp. 292-294.
6. A. Ray, A. Surana and S. Phoha, "A Language Measure for Supervisory Control," ***Applied Mathematics Letters***, Vol. 16, September 2003, pp. 985-991.
7. B. S. Hong, A. Ray and V. Yang, "Output Feedback Linear Parameter Varying (LPV) L2-gain Control," ***ASME Journal of Dynamic Systems, Measurement, and Control***, Vol. 125, No. 3, September 2003, pp. 485-489.
8. E. Keller and A. Ray, "Real-time Nondestructive Evaluation of Mechanical Structures," ***Structural Health Monitoring,*** Vol. 2, No. 3, September 2003, pp. 191-203.
9. A. Ray and S. Phoha, "Calibration and Estimation of Redundant Signals for Real-time Monitoring and Control," ***Signal Processing***, Vol. 83, No.12, December 2003, pp. 2593-2605.
10. A. Ray and S. Phoha, "Signed Real Measure of Regular Languages for Discrete-Event Automata," ***International Journal of Control***, Vol. 76, No. 18, December 2003, pp. 1800-1808.
11. J. Fu, A. Ray and C. M. Lagoa, "Unconstrained Optimal Control of Regular Languages," ***Automatica***, Vol. 40, No. 4, April 2004, pp. 639-646.
12. A. Surana and A. Ray, "Measure of Regular Languages," ***Demonstratio Mathematica***, Vol. 37, No. 2, June 2004, pp. 485-503.
13. A. Ray, "Symbolic Dynamic Analysis of Complex Systems for Anomaly Detection," ***Signal Processing***, Vol. 84, No. 7, July 2004, pp. 1115-1130.
14. A. Ray, J. Fu and C.M. Lagoa, "Optimal Supervisory Control of Finite State Automata," ***International Journal of Control***, Vol. 77, No. 12, August 2004, pp. 1083-1100.
15. X. Wang and A. Ray, "A Language Measure for Performance Evaluation of Discrete Event Supervisory Control Systems," ***Applied Mathematical Modelling***, Vol. 28, No. 9, September 2004, pp. 817-833.
16. V. Phoha, A. Nadgar, A. Ray, S. Phoha and V. Jain, "Supervisory Control of Software Systems," ***IEEE Transactions on Computers***, Vol. 53, No. 9, September 2004, pp. 1187-1199.
17. A. Ray, "Stochastic Measure of Fatigue Crack Damage for Health Monitoring of Ductile Alloy Structures," ***Structural Health Monitoring***, Vol. 3, No. 3, September 2004, pp. 245-263.
18. A. Petruzzi, S. Chin, K. Ivanov, A. Ray and F-B. Cheung, "Development of Methodology for Early Detection of BWR Instabilities," ***Trans. of American Nuclear Soc.***, Vol. 91, November 2004, pp. 902-904.

19. X. Wang, A. Ray, P. Lee and J. Fu, “Optimal Control of Robot Behavior Using Language Measure,” ***International Journal of Vehicle Autonomous Systems***, Vol. 2, Nos. 3/4, 2004, pp. 147-167.

20. A. Ray, J. Fu and C. Lagoa, “Optimal Control of Regular Languages with Event Disabling Cost,” ***Demonstratio Mathematica***, Vol. 37, No. 4, 2004, pp. 991-1013.

21. X. Wang and A. Ray, “A Language Measure for Performance Quantification of Discrete Event Supervisory Control Systems,” ***Bulletin of Cal. Math. Soc.***, Vol. 97, No. 1, 2005, pp. 59-68.

22. W. Xie, Y. Hong, and K. S. Trivedi, “Analysis of a Two-Level Software Rejuvenation Policy,” ***Reliability Engineering and System Safety***, Vol. 87, No. 7, 2005, pp. 13-22.

23. X. Wang, A. Ray and A. Khatkhate, “On-line Identification of Language Measure Parameters for Discrete Event Supervisory Control,” ***Applied Mathematical Modelling***, Vol. 29, No. 6, June 2005, pp. 597-613.

24. J. F. Horn, D. K. Tolani, C. M. Lagoa, Q. Wang and A. Ray, “Probabilistic Robust Control of Rotorcraft,” ***Control Engineering Practice***, Vol. 13, No. 8, August 2005, pp. 1037-1046.

25. C. Lagoa, J. Fu and A. Ray, “Robust Optimal Control of Regular Languages,” ***Automatica***, Vol. 41, No. 8, August 2005, pp. 1339-1445.

26. A. Ray, “Signed Real Measure of Regular languages for Discrete Event Supervisory Control,” ***International Journal of Control***, Vol. 78, No. 12, August 2005, pp. 949-967.

27. Y. Sheng, V. V. Phoha and S. M. Rovnyak, “A Parallel Decision Tree-based Method for User Authentication Based on Keystroke Patterns,” ***IEEE Transactions on System, Man, and Cybernetics***, Vol. 35, No. 4, August 2005, pp. 826-833.

28. S. Chin, A. Ray and V. Rajagopalan, “Symbolic Time Series Analysis for Anomaly Detection: A Comparative Evaluation,” ***Signal Processing***, Vol. 85, No. 9, September 2005, pp. 1859-1868.

29. I. Chattopadhyay and A. Ray, “A Complex Measure for Linear Grammars,” ***Demonstratio Mathematica***, Vol. 38, No. 3, 2005, pp. 761-775.

30. N. Scafetta, A. Ray and B. J. West, “Correlation Regimes in Fluctuations of Fatigue Crack Growth,” ***Physica A***, Vol. 59, No. 3, 2006, pp. 1-23.

31. D. K. Tolani, M. Yasar, A. Ray and V. Yang, “Anomaly Detection in Aircraft Gas Turbine Engines,” ***AIAA Journal of Aerospace Computing, Information, and Communication***, Vol. 3, No. 2, 2006, pp. 44-51.

32. D. K. Tolani, A. Ray and J. F. Horn, “Integrated Decision and Control of Human-Engineered Complex Systems,” ***International Journal of General Systems***, Vol. 35, No. 3, June 2006, pp. 275-294.

33. V. Rajagopalan and A. Ray, “Wavelet Space Partitioning for Symbolic Time Series Analysis,” ***Chinese Physics Letters***, Vol. 23, No. , July 2006, pp. 1951-1954.

34. S. Gupta, A. Ray and E. Keller, “Symbolic Time Series Analysis of Ultrasonic Signals for Fatigue Damage Monitoring in Polycrystalline Alloys,” ***Measurement Science and Technology***, Vo. 17, Issue 7, July 2006, pp. 1963-1973.

35. A. Khatkhate, A. Ray, E. Keller, S. Gupta and S. Chin, “Symbolic Time Series Analysis for Anomaly Detection in Mechanical Systems,” ***IEEE/ASME Trans. on Mechatronics***, Vol. 11, No. 4, August 2006, pp. 439-447.

36. S. Gupta, A. Ray and A. Mukhopadhyay, “Anomaly Detection in Thermal Pulse Combustors,” ***Proceedings of the I Mech E Part I Journal of Systems & Control Engineering,*** Vol. 220, No. 5, August 2006, pp. 339-351.

37. I. Chattopadhyay and A. Ray, “A Language Measure for Partially Observed Discrete Event Systems,” ***International Journal of Control***, Vol. 79, No. 9, September 2006, pp. 1074-1086.

38. I. Chattopadhyay and A. Ray, “Renormalized Measure of Regular Languages,” ***International Journal of Control***, Vol. 79, No. 9, September 2006, pp. 1107-1117.

39. S. Gupta, A. Khatkhate, A. Ray and E. Keller, “Identification of Statistical Patterns in Complex Systems via Symbolic Time Series Analysis,” ***ISA Transactions***, Vol. 45, No. 4, October 2006, pp. 477-490.

40. V. Rajagopalan and A. Ray, “Symbolic Time Series Analysis via Wavelet-based Partitioning,” ***Signal Processing***, Vol. 86, No. 11, November 2006, pp. 3309-3320.

41. M. Yasar and A. Ray, “Hierarchical Control of Aircraft Propulsion Systems: Discrete Event Supervisor Approach,” ***Control Engineering Practice***, Vol. 15, No. 2, February 2007, pp. 149-162.

42. S. Gupta, A. Ray and E. Keller, “Symbolic Time Series Analysis of Ultrasonic Data for Early Detection of Fatigue Damage,” ***Mechanical Systems and Signal Processing,*** Vol. 21, No. 2, February 2007, pp. 866-884.

43. S. R. Gaddam, V. V. Phoha, and K. S. Balagani, “K-Means+ID3: A Novel Method for Supervised Anomaly Detection by Cascading k-Means Clustering and ID3 Decision Tree Learning Methods,” ***IEEE Transactions on Knowledge and Data Engineering***, Vol. 19, No. 3, March 2007, pp. 345-354.

44. A. Khatkhate, A. Ray and E. Keller, “Modelling and System Identification of an Experimental Apparatus for Anomaly Detection in Mechanical Systems,” ***Applied Mathematical Modelling***, Vol. 31, No. 4, April 2007, pp. 734-748.

45. I. Chattopadhyay and A. Ray, “Generalized Language Measure Families of Probabilistic Finite State Systems,” ***International Journal of Control***, Vol. 80, No. 5, May 2007, pp. 789-799.

46. S. Gupta and A. Ray, “Real-Time Fatigue Life Estimation in Mechanical Structures,” ***Measurement Science and Technology***, vol. 18, May 2007, pp. 1947-1957.

47. S. Gupta, A. Ray and E. Keller, “Fatigue Damage Monitoring by Ultrasonic Measurements: A Symbolic Time Series Analysis Approach,” ***International Journal of Fatigue***, Vol. 29, No. 6, June 2007, pp. 1100-1114.

48. I. Chattopadhyay and A. Ray, “Language-Measure-Theoretic Optimal Control of Probabilistic Finite-State Systems,” ***International Journal of Control***, in press.

49. V. Rajagopalan, A. Ray, R. Samsi and J. Mayer, “Pattern Identification in Dynamical Systems via Symbolic Time Series Analysis,” ***Pattern Recognition***, in press.

50. R. Patankar, V. Rajagopalan and A. Ray, “Fault Precursor Detection in Complex Electrical Systems Using Symbolic Dynamics,” ***International Journal of Signal and Image Processing Engineering***, in press.

51. A. Khatkhate, S. Gupta, A. Ray and R. Patankar, “Anomaly Detection in Flexible Mechanical Couplings via Symbolic Time Series Analysis,” ***Journal of Sound and Vibration***, in press.

(b) Papers published in non-peer-reviewed journals or in refereed conference proceedings

1. E. Keller and A. Ray, “Real-Time Nondestructive Evaluation of Airframe Structures for Health Monitoring and Residual Life Prediction,” ***20th Digital Avionics Systems Conference***, West Palm Beach, Florida, October 2001.

2. A. Ray and S. Phoha, “Detection and Identification of Emerging Faults,” ***IEEE Conference on Decision and Control (CDC)***, Orlando, Florida, December 2001.

3. X. Wang and A. Ray, “Signed Real Measure of Regular Languages,” ***American Control Conference***, Anchorage, Alaska, May 2002.

4. J. Fu, A. Ray and J. Spare, “Load Scheduling and Health Management of Electric Power Generation Systems,” ***American Control Conference***, Anchorage, Alaska, May 2002.

5. A. Ray and S. Phoha, “Calibration and Estimation of Redundant Signals,” ***American Control Conference***, Anchorage, Alaska, May 2002.

6. Y. Hong, D. Chen, L. Li, and K. S. Trivedi, “Closed Loop Design for Software Rejuvenation,” ***Workshop on Self-Healing, Adaptive, and Self-Managed Systems***, June, 2002.

7. D. Chen, S. Garg, C. Kintala and K. S. Trivedi, “Dependability Enhancements for Wireless Local Area Networks with Redundancy Techniques,” ***International Conference on Dependable Systems and Networks, Performance and Dependability Symposium***, San Francisco, CA, June 2003.

8. A. Lou, “Fault Manifestation Model for Predicting Anomalous System Behavior,” ***The International Conference on Dependable Systems and Networks***, Washington D.C., June 2002.

9. A. Ray and S. Phoha, “A Language Measure for Discrete-Event Automata,” ***International Federation of Automatic Control (IFAC) World Congress b’02***, Barcelona, Spain, July 2002.

10. A. Ray and S. Phoha, “Fault Detection and Identification via Multi-level Hypotheses Testing,” ***International Federation of Automatic Control (IFAC) World Congress b’02***, Barcelona, Spain, July 2002.

11. J. Metzner and U. Tuntoolavest, “Pulse Time Hopping for Multiaccess Communications with a Concatenated Code,” ***IEEE International Midwest Symposium on Circuits and Systems***, Oklahoma State Univ., August 2002.

12. M. Li and J. Metzner, “Reliable Satellite Multicast with Assistance of Terrestrial Communications,” ***IEEE International Midwest Symposium on Circuits and Systems***, Oklahoma State Univ., August 2002.

13. J. Metzner, “Burst Erasure Correction to Improve the Efficiency of Broadband CSMA/CD,” ***IEEE International Midwest Symposium on Circuits and Systems***, Oklahoma State Univ., August 2002.

14. J. Dorsey and D. P. Siewiorek, “Defect Distribution for Wearable System Design,” ***International Symposium on Wearable Computers***, Seattle, WA, October 2002.

15. J. Fu, A. Ray and C. Lagoa, “Unconstrained Optimal Control of Regular Languages,” ***IEEE Conference on Decision and Control (CDC)***, Las Vegas, Nevada, December 2002.

16. Y. Hong, D.-Y. Chen, and K. S. Trivedi, “Failure Mitigation for Quality of Service of Wireless Networks,” ***IEEE Conference on Decision and Control (CDC)***, Las Vegas, Nevada, December 2002.

17. J. Dorsey and D. P. Siewiorek, “The Design of Wearable Systems: A Shift in Development Effort,” ***International Conference on Dependable Systems and Networks***, San Francisco, CA, June 2003.

18. D. Friedlander, I. Chattopadhyay, A. Ray, S. Phoha and N. Jacobson, “Anomaly Prediction in Mechanical Systems Using Symbolic Dynamics,” ***American Control Conference***, Denver, Colorado, June 2003.

19. Y. Hong, K.S. Trivedi, A. Ray and S. Phoha, “Software Performance Analysis Using a Language Measure,” ***American Control Conference***, Denver, Colorado, June 2003.

20. J. Fu, A. Ray and C. Lagoa, “Optimal Control of Regular Languages with Disabling Cost,” ***American Control Conference***, Denver, Colorado, June 2003.

21. X. Wang, A. Ray, S. Phoha and J. Liu, “J-DES: A Graphical Interactive Package for Analysis and Synthesis of Discrete Event Systems,” ***American Control Conference***, Denver, Colorado, June 2003.

22. T. Ortogero, A. Ray and S. Phoha, “Validation of a Language Measure for Discrete-Event Supervisory Control,” ***American Control Conference***, Denver, Colorado, June 2003.

23. V. Phoha, A. Nadgar, A. Ray, J. Fu and S. Phoha, “Supervisory Control of Software Systems for Fault Mitigation,” ***American Control Conference***, Denver, Colorado, June 2003.

24. J.J. Metzner, S. Chin and A. Ray, “Control of Communication Networks for Multiple Mobile Platforms,” ***American Control Conference***, Denver, Colorado, June 2003.

25. J. F. Horn, D. K. Tolani, C. M. Lagoa, Q. Wang and A. Ray, “Reliable Operation of Rotorcraft Using Probabilistic Robust Control,” ***5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes***, Washington, D.C., June 2003.

26. V. Phoha, X. Xu, A. Ray, J. Fu and S. Phoha, “Supervisory Control of Malicious Executables,” ***5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes***, Washington, D.C., June 2003.

27. A. Surana, A. Ray and S. Chin,” Anomaly Prediction in Complex Systems,” ***5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes***, Washington, D.C., June 2003.

28. C. Delfino, A. Ray, K.N. Ivanov, F-B. Cheung, “A New Concept for Early Detection of BWR Instabilities,” ***American Nuclear Society Meeting***, *San Diego*, CA, June 2003.

29. X. Wang, P. Lee, A. Ray and S. Phoha, “A Behavior-based Collaborative Multi-Agent System,” ***IEEE Conference on System, Man and Cybernetics***, Washington, D.C., October 2003.

30. J. Fu, C. Lagoa and A. Ray, “Robust Optimal Control of Regular Languages with Event Cost Uncertainties,” ***IEEE Conference on Decision and Control (CDC)***, Maui, Hawaii, December 2003.

31. X. Wang, A. Ray and A. Khatkhate, “On-line Identification of Language Measure Parameters for Discrete Event Supervisory Control,” ***IEEE Conference on Decision and Control (CDC)***, Maui, Hawaii, December 2003.

32. A. Surana and A. Ray, “Signed Real Measure of Regular Languages,” ***IEEE Conference on Decision and Control (CDC)***, Maui, Hawaii, December 2003.

33. W. Xie, Y. Hong, and K. S. Trivedi, “Software Rejuvenation Policies for Cluster Systems under Varying Workloads,” ***Pacific Rim International Symposium on Dependable Computing***, Tahiti, French Polynesia, March 2004.

34. D. Tolani, J. Horn, A. Ray and J. Chen, “Hierarchical Control of Future Generation Aircraft,” ***American Control Conference***, Boston, MA, June-July 2004.

35. I. Chattopadhyay, A. Ray and X. Wang, “A Complex Measure of Formal Languages Generated by Linear grammars,” ***American Control Conference***, Boston, MA, June-July 2004.

36. X. Wang, J. Fu, P. Lee and A. Ray, “Robot Behavioral Selection Using a Discrete-Event Language Measure,” ***American Control Conference***, Boston, MA, June-July 2004.

37. A. Khatkhate, E. Keller, A. Ray and S. Chin, “Early Detection of Fatigue Crack Anomaly: A Symbolic Dynamic Approach,” ***American Control Conference***, Boston, MA, June-July 2004.

38. J. Fu, M. Yasar and A. Ray, “Optimal Discrete Event Supervisory Control of Aircraft Gas Turbine Engines,” ***American Control Conference, Boston***, MA, June-July 2004.

39. G. K. Kuchimanachi, V. V. Phoha, K. S. Balagani and S. R. Gaddam, “Dimension Reduction Using Feature Extraction Methods for Real-time Misuse Detection Systems,” ***IEEE Information Assurance Workshop***, West Point Military Academy, New York, 2004.

40. D. P. Siewiorek, R. A. Maxion and P. Narasimhan, “Experimental Research in Dependable Computing at Carnegie Mellon University” ***IFIP 18th World Computer Congress***, Toulouse, France, August 2004.

41. A. Petruzzi, S. Chin, K. Ivanov, A. Ray and F-B. Cheung, “Development of Methodology for Early Detection of BWR Instabilities,” ***American Nuclear Society Meeting***, Washington, D.C., November 2004.

42. V. Rajagopalan, R. Samsi, A. Ray, J. Meyer and C. Constantino, “A Symbolic Dynamics Approach for Early Detection of Slowly Evolving Faults in Nonlinear Systems,” ***IASTED Conference***, Orlando, Florida, November 2004.

43. I. Chattopadhyay and A. Ray, “A Language Measure for Partially Observed Discrete Event Supervisory Control Systems,” ***IEEE Conference on Decision and Control (CDC)***, Paradise Island, Bahamas, December 2004.

44. X. Wang, I. Chattopadhyay and A. Ray, “Probabilistic Fault Diagnosis in Discrete Event Systems,” ***IEEE Conference on Decision and Control (CDC)***, Paradise Island, Bahamas, December 2004.

45. D. Tolani, M. Yasar, S. Chin and A. Ray, “Anomaly Detection for Health Management of Aircraft Gas Turbine Engines,” ***American Control Conference***, Portland, OR, June 2005.

46. M.Yasar, D. Tolani and A. Ray, “Optimal Supervisory Control of Aircraft Propulsion,” ***American Control Conference***, Portland, OR, June 2005.

47. S. Bhatnagar, V. Rajagopalan and A. Ray, “Incipient Fault Detection in Mechanical Power Systems,” ***American Control Conference***, Portland, OR, June 2005.

48. R. Samsi, V. Rajagopalan, J. Meyer and A. Ray, “Early Detection of Voltage Imbalances in Induction Motors,” ***American Control Conference***, Portland, OR, June 2005.

49. S. Gupta, A. Ray, E. Keller and S. Chin, “Online Detection of Fatigue Failure via Symbolic Time Series Analysis,” ***American Control Conference***, Portland, OR, June 2005.

50. X. Wang, G. Mallapragada and A. Ray, “Language-measure-based Supervisory Control of a Mobile Robot,” ***American Control Conference***, Portland, OR, June 2005.

51. C. Lagoa, J. Chen and A. Ray, “On the Design of Discrete-Time Fixed-Order Controllers for Persistent Disturbance Rejection,” ***16th IFAC World Congress***, Prague, Czech Republic, July 2005.

52. D. K. Tolani, J. F. Horn, M. Yasar and A. Ray, “Hierarchical Control of Rotorcraft for Enhanced Performance and Structural Durability,” ***AIAA Guidance, Navigation, and Control Conference***, San Francisco, August 2005.

53. D. O. Bridges, J. F. Horn and A. Ray, “Model-Following Control of a Military Helicopter with Damage Mitigation,” ***AIAA Guidance, Navigation, and Control Conference***, San Francisco, August 2005.

54. J. Chen, C. M. Lagoa, J. F. Horn and A. Ray, “Output Bounded Switching Control for Future Generation Rotorcraft,” ***AIAA Guidance, Navigation, and Control Conference***, San Francisco, August 2005.

55. I. Chattopadhyay, S. Chakraborty and A. Ray, “Intelligent Navigation in Space under Supervisory Control,” ***AIAA 5th Aviation, Technology Integration, and Operations (ATIO) Conference***, Arlington, VA, September 2005.

56. V. Rajagopalan and A. Ray, “Wavelet-based Space Partitioning for Symbolic Time Series Analysis,” ***IEEE Conference on Decision and Control (CDC) and European Control Conference (ECC)***, Seville, Spain, December 2005.

57. V. V. Phoha, “A Simple Real Time Executive tinyexec as a Teaching Tool in an Operating System Class,” ***Third Annual Mid-South College Computing Conference***, Oxford, MS, 2005.

58. R. Samsi, V. Rajagopalan and A. Ray, “Wavelet-based Symbolic Analysis for Fault Detection,” ***American Control Conference***, Minneapolis, MN, June 2006.

59. A. Khatkhate, S. Gupta, A. Ray and E. Keller, “Life Extending Control of Mechanical Systems using Symbolic Time Series Analysis,” ***American Control Conference***, Minneapolis, MN, June 2006.

60. M. Yasar, J. F. Horn and A. Ray, “Effects of Supervisory Decisions on Nonlinear Aircraft Dynamics,” ***American Control Conference***, Minneapolis, MN, June 2006.

61. I. Chattopadhyay and A. Ray, “Renormalized Measure of Regular Languages,” ***WODES 2006***, Ann Arbor, MI, July 2006.

62. I. Chattopadhyay and A. Ray, “Generalized Language Measure of Finite State Logical Systems,” ***WODES 2006***, Ann Arbor, MI, July 2006.

63. K. S. Balagani and V. V. Phoha, “A Novel Sliding-window Algorithm for Fast Detection of SYN Flooding Attacks on Distributed Sensor Networks,” ***International Innovations and Real-time Applications of Distributed Sensor Networks (DSN) Symposium***, Washington D.C., October 2006.

64. G. Mallapragada, I. Chattopadhyay and A. Ray, "Autonomous Navigation of Mobile Robots Using Optimal Control of Finite State Automata," ***IEEE Conference on Decision and Control (CDC)***, San Diego, CA, December 2006.

65. R. Samsi, J. Mayer and A. Ray, "Broken Rotor Bar Detection Using Symbolic Wavelet Analysis," ***IEEE Conference on Decision and Control (CDC)***, San Diego, CA, December 2006.

66. V. V. Phoha and S. P. Phoha, "Situation-Aware Software Engineering for Sensor Networks," The ***IEEE International Conference on Communication Systems Software and Middleware***, Bangalore, India, January 2007.

67. S. Joshi and V. V. Phoha, "Competition between SOM Clusters to Model User Authentication System in Computer Networks," ***IEEE International Conference on Communication Systems Software and Middleware***, Bangalore, India, January 2007.

68. Y. B. Reddy and V. V. Phoha, "Genetic Algorithm Approach for Resource Allocation in Multi-User OFDM Systems," ***IEEE International Conference on Communication Systems Software and Middleware***, Bangalore, India, January 2007.

69. S. Phoha and A. Ray, "Information Driven Design of Urban Sensor Networks," ***IEEE International Conference on Networking, Sensing and Control***, London, UK, April 2007.

70. K. S. Balagani, V. V. Phoha, and G. Kuchimanchi, "A Divergence-measure Based Classification Method for Detecting Anomalies in Network Traffic," ***IEEE International Conference on Networking, Sensing, and Control***, London, UK, April 2007.

71. A. Subbu, E. Keller and A. Ray, "A symbolic dynamics based approach to pattern recognition in image sequences," ***International Conference on Image Processing, Computer Vision, and Pattern Recognition***, Las Vegas, Nevada, June 2007.

72. S. Gupta and A. Ray, "Estimation of Fatigue Life Using Ultrasonic Sensing: A Symbolic Dynamics Approach," ***American Control Conference***, New York, NY, July 2007.

73. A. Srivastav, A. Ray and S. Gupta, "Irreversibility-Based Measure of Slowly Evolving Anomalies," ***American Control Conference***, New York, NY, July 2007.

74. I. Chattopadhyay and A. Ray, "Generalized Projections & Decidability of State Determinacy in Finite State Automata," ***American Control Conference***, New York, NY, July 2007.

75. R. P. Patankar, V. Rajagopalan, D. Tolani and A. Ray, "Prognosis of Failure Precursor in Complex Electrical Systems Using Symbolic Dynamics," ***American Control Conference***, New York, NY, July 2007.

(c) Manuscripts submitted but not yet published.

1. M. Yasar, A. Ray and J. F. Horn, "A Comprehensive Control Strategy for Integrated Flight/Propulsion Systems", ***Control Engineering Practice***, under review.

2. M. Yasar, S. Sarkar, S. Gupta and A. Ray, "Fault Diagnosis and Isolation in Aircraft Gas Turbine Engines: Part I - The Underlying Concept," ***AIAA Journal of Propulsion and Power***, under review.

3. M. Yasar, S. Sarkar, S. Gupta and A. Ray, "Fault Diagnosis and Isolation in Aircraft Gas Turbine Engines: Part II - Validation on a Simulation Test Bed," ***AIAA Journal of Propulsion and Power***, under review.

4. A. Srivastav, A. Ray and S. Gupta, "An Information-theoretic Measure for Anomaly Detection in Complex Dynamical Systems," ***Mechanical Systems and Signal Processing***, under review.

5. C. Rao, S. Sarkar, M. Yasar and A. Ray, "Comparative Evaluation of Symbolic Dynamic Filtering for Detection of Anomaly Patterns," ***Pattern Recognition***, under review.

6. I. Chattopadhyay and A. Ray, "Generalized Projections in Finite State Automata," ***IEEE Transactions on Automatic Control***, under review.

7. V. Rajagopalan, S. Chakraborty and A. Ray, "Estimation of Slowly-Varying Parameters in Nonlinear Systems via Symbolic Dynamic Filtering," ***Signal Processing***, under review.

8. S. Chakraborty, S. Gupta, A. Ray and A. Mukhopadhyay, "Dynamic Data-Driven Detection and Estimation of Faults in Thermal Pulse Combustors," ***ASME Journal of Dynamic Systems, Measurement, and Control***, under review.

9. G. Mallapragada, I. Chattopadhyay and A. Ray," Automated Behavior Recognition in Mobile Robots Using Symbolic Dynamic Filtering," ***Robotics and Autonomous Systems***, under review.

(d) Book and book chapters

1. A. Ray, V. Phoha and S. Phoha, ***Quantitative Measure for Discrete Event Supervisory Control: Theory and Applications***, Springer, New York, 2005. **ISBN 0-387-02108-1**.

2. A. Ray and S. Phoha, "Sensor Calibration and Estimation for Real-Time Monitoring and Control," ***Frontiers In Distributed Sensor Networks***, CRC Press, Boca Raton, Florida, eds. S.S. Iyenger and R. Brooks, 2005, pp. 381-398.

3. V.V. Phoha, S. Phoha, A. Ray, K.S. Balagani, A.U. Nadgar and R. Varanasi, "Embedded Soft Sensing for Anomaly Detection in Mobile Robotic Networks," ***Sensor Network Operations***, IEEE Press/Wiley, eds. S. Phoha, T. LaPorta and C. Griffin, Chapter 10, 2006, pp. 609-629.

4. S. Gupta and A. Ray, "Symbolic Dynamic Filtering for Data-driven Pattern Recognition," ***Pattern Recognition Research Horizons***, ed., F. Columbus, Nova Science Publishers, Hauppage, NY, in press.

(e) Technical reports

1. First Year Interim Report for *2001 MURI on Characterization and Mitigation of Service Failures in Complex Dynamic Systems*, funding number **C-DAAD19-01-1-0646**, March 2002.

2. Second Year Interim Report for *2001 MURI on Characterization and Mitigation of Service Failures in Complex Dynamic Systems*, funding number **C-DAAD19-01-1-0646**, March 2003.

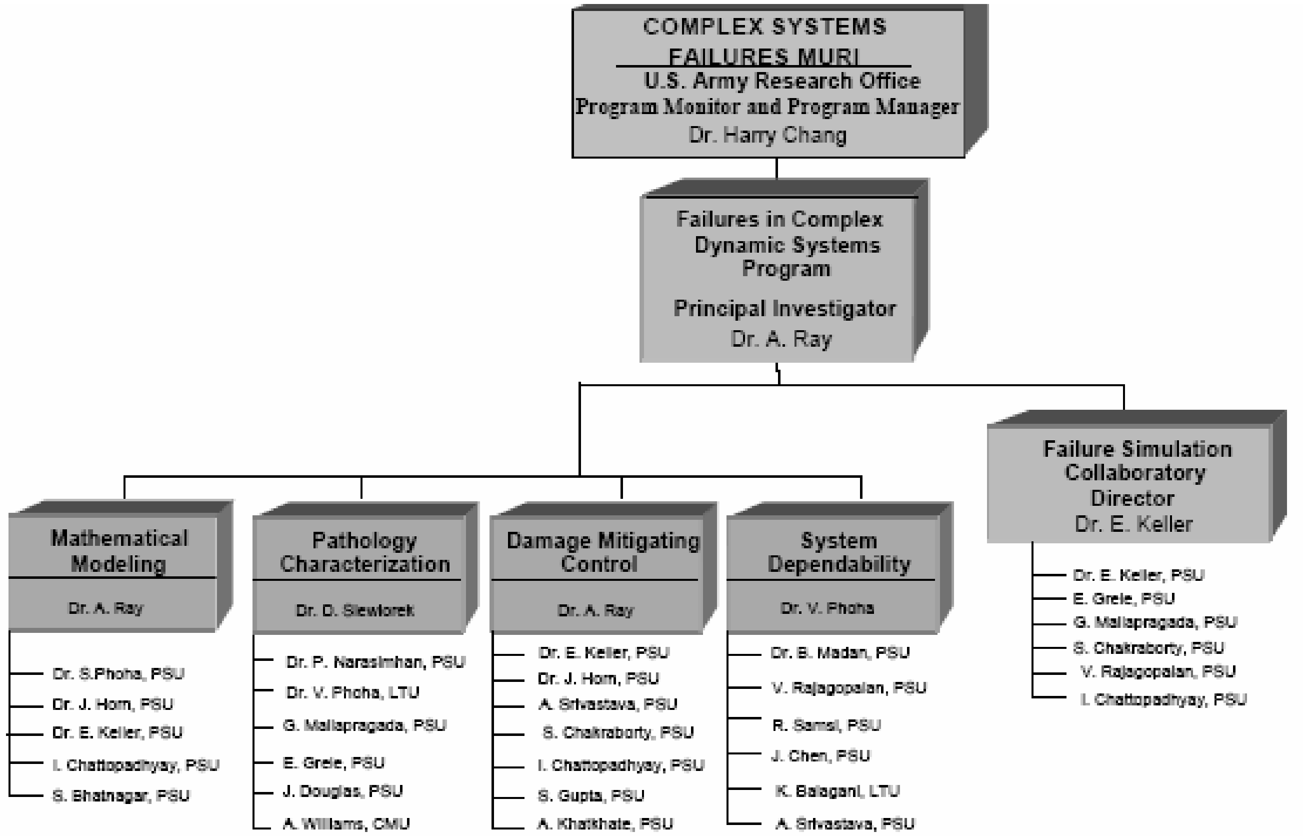
3. Third Year Interim Report for *2001 MURI on Characterization and Mitigation of Service Failures in Complex Dynamic Systems*, funding number **C-DAAD19-01-1-0646**, March 2004.

4. Third Year Interim Report for *2001 MURI on Characterization and Mitigation of Service Failures in Complex Dynamic Systems*, funding number **C-DAAD19-01-1-0646**, August 2004.

5. Fourth Year Interim Report for *2001 MURI on Characterization and Mitigation of Service Failures in Complex Dynamic Systems*, funding number **C-DAAD19-01-1-0646**, August 2005.

6. Fifth Year Interim Report for *2001 MURI on Characterization and Mitigation of Service Failures in Complex Dynamic Systems*, funding number **C-DAAD19-01-1-0646**, August 2006.

LIST OF PARTICIPATING SCIENTIFIC PERSONNEL



Degrees:

- Dr. Xi Wang, M.S. in Electrical Eng., Ph.D. in Mechanical Eng., PSU, 2003
- Dr. Jinbo Fu, M.S. in Electrical Eng., M.S. in Mechanical Eng., Ph.D. in Mechanical Eng., PSU, 2003
- Dr. Shin Chin, Ph.D. in Electrical Eng., PSU, 2004
- Dr. Devendra K. Tolani, M.S. in Electrical Eng., M.S. in Mechanical Eng., Ph.D. in Mechanical Eng., PSU, 2005
- Dr. Jialing Chen, Ph.D. in Electrical Eng., PSU, 2005
- Dr. Weihua Song, Ph.D. in Computer Science, LTU, 2005
- Dr. Shalabh Gupta, M.S. in Mechanical Eng., M.S. in Electrical Eng., Ph.D. in Mechanical Eng., PSU, 2006
- Dr. Amol Khatkhate, M.S. in Mechanical Eng., Ph.D. in Mechanical Eng., PSU, 2006
- Dr. Ishanu Chattopadhyay, M.A. in Mathematics, M.S. in Mechanical Eng., Ph.D. in Mechanical Eng., PSU, 2006
- Dr. Xin Xu, M.S. in Computer Science, Ph.D. in Computer Science, LTU, 2006
- Dr. Rohan Samsi, M.S. in Mechanical Eng., Ph.D. in Electrical Eng., PSU, 2007
- Dr. Venkatesh Rajagopalan, Ph.D. in Electrical Eng., PSU, 2007
- Dr. Murat Yasar, M.S. in Electrical Eng., M.S. in Mechanical Eng., Ph.D. in Mechanical Eng., PSU, 2007

- Vijay Jain, M.S. in Computer Science, LTU
- Amit Nadgar, M.S. in Computer Science, LTU
- S. Vyurru, M.S. in Computer Science, LTU
- P. Kumar, M.S. in Computer Science, LTU
- S. Gaddam, M.S. in Computer Science, LTU
- G. K. Kuchimanachi , M.S. in Computer Science, LTU
- Sunil Babu, M.S. in Computer Science, LTU
- Kesari Mishra, M.S. in Electrical and Computer Eng., Duke U
- L. Li, M.S. in Mechanical Eng., Duke U
- Andrew Williams, M.S. in Computer Science, CMU
- Anirban Bagchi, M.S. in Information Systems Management, CMU
- Amit Surana, M.S. in Mechanical Eng, M.A. in Mathematics, PSU
- R. Scarmach, M.S. in Mechanical Engineering, PSU
- Raul Goel, M.S. in Mechanical Engineering, PSU
- Aparna Subbu, M.S. in Mechanical Eng., PSU
- Saurabh Bhatnagar, M.S. in Electrical Eng., M.S. in Mechanical Eng., PSU
- Goutham Mallapragada, M.S. in Electrical Eng., M.S. in Mechanical Eng., PSU
- Subhadeep Chakraborty, M.S. in Electrical Eng., M.S. in Mechanical Eng., PSU
- Abhishek Srivastav, M.S. in Electrical Eng., M.S. in Mechanical Eng., PSU
- Chinmay R. Rao, M.S. in Electrical Eng., M.S. in Mechanical Eng., PSU

<u>Awards / Honors:</u>

- Dr. Asok Ray (PSU Faculty)
  - 2002 IEEE Fellowship award
  - 2002 Premier Research award of the Pennsylvania State University, College of Engineering
  - Best Paper Award for ASME Journal of Dynamic Systems, Measurements, and Control
  - Distinguished Professorship in The College of Engineering of Pennsylvania State University
  - Academia member of the Governing Board of the Process Control Forum of American Automatic Control Council (AACC) in July 2006
  - Most cited paper in Signal Processing, 2004-2006

- Dr. Daniel P. Siewiorek (CMU Faculty)
  - University Professorship
  - Allen Newell Award for Research Excellence
  - Editorial Board for the Launch of a New IEEE Computer Society Publication – Pervasive Computing Magazine. A new professional publication started in 2002
  - Keynote talk “Pervasive and Context Aware Computing” at Asian Pacific Computer Human Interaction Conference, November 2, 2002
  - General Chairman for a new ACM Conference on Mobile Systems: BioSys, May 2003
  - Chair of IEEE Computer Society Technical Committee on Wearable Information Systems

  - General chair of an IEEE Conference honoring Ed McCluskey
  - General chair of an IFIP Working Group 10.4 Workshop on Human Error
  - Member of two National Research Council Panels (Naval Science Board on ForceNet and Board on Army Science and Technology on Power for the Dismounted Soldier), 2004
  - Best Poster award at Microsystems Education Conference

- Priya Narasimhan (CMU Faculty)
  - IBM Faculty Partnership Award, 2002
  - Program Chair, Workshop on Foundations of Middleware Technologies, Irvine, CA, November 2002
  - National Science Foundation CAREER Award for proposed research on "Integrated Real-Time and Fault-Tolerance Support for Middleware Applications", 2003-2008
  - Program Chair, IEEE Workshop on Dependable Middleware-Based Systems, Mexico, 2003
  - Program Chair, IEEE Workshop on Object-oriented Real-time Dependable Systems, Mexico, 2003

- Dr. Kishor Trivedi (Duke University Faculty)
  - Poonma and Prabhu Goel Chair as a Visiting Professor to IIT Kanpur (Aug 2002 – May 2003)
  - Fulbrignt visiting lecturer award to India (November 2002-May 2003)

- Dr. Shashi Phoha (PSU Faculty) – 2005 IEEE Computer Society Technical Achievement award

- Mr. Andrew Williams (CMU Graduate Student) – The prestigious Department of Homeland Security fellowship starting September 2005

REPORT OF INVENTIONS

1. **V. V. Phoha, Sunil Babu, Asok Ray, Shashi Phoha, "System and Method for Classifying Regions of Keystroke Density with Neural Network,"** ***Technology licensed to BioPassword Inc., Washington*.** Patent filed through Blakely Sokoloff Taylor & Zafman, LLP.
2. **V. V. Phoha, "Choice of Sample Size for Keystroke Authentication,"** ***Technology licensed to BioPassword Inc., Washington*.** Patent filed through Blakely Sokoloff Taylor & Zafman, LLP.

PSU Invention Disclosure No. 2003-0780.

- Technology transferred to and licensed with BioPassword Inc., Issaquah, WA 98027.
- The system is incorporated in version of their Enterprise and Network authentication system.
- First ever direct income from an invention for Louisiana Tech University
- Appeared in the national news

Background: The multidisciplinary research on anomaly detection and failure pattern discovery in Electromechanical Systems and Networked Robotic systems, conducted under the ARO grant DAAD19-01-06046, has been extended to *Computer User Login Authentication*. The objective is to permit authorized entry and inhibit unauthorized access into computer systems and networks, based on the key stroke typing of individual users. This neural-network-based authentication algorithm makes a suboptimal trade-off between the probabilities of correct authorization and false through classification of the Keystroke-typing patterns as *normal* or *anomalous*.

Key Features: The neural network architecture in key-stroke pattern classification has two types of connections: (i) weight vector **W**; and (ii) dispersion vector **V**. In the learning phase, **W** adapts to the users' keystroke exemplars, and **V** adapts to dispersion (e.g., standard deviation) of the users' keystrokes. The algorithm consists of three phases: training, validation, and testing. The authentication system learns the vectors **W** and **V** in the training phase. The system parameters, Scale Factor (SF) and Percentage Success (PS), are adjusted during the validation phase. In the testing and actual-use phase, correct classification results in fine tuning **W** and **V**, thereby adapting to the users' typing pattern. The weight vector **W** and the dispersion vector **V** adapt to the changing typing patterns of the users during regular use.

Contributions: The significant contributions of the neural-network-based authentication algorithm are outlined below:

(1) A novel neural network architecture to build authentication systems

(2) Self adaptation to changes in typing patterns without re-training the network

(3) Robustness of network performance connections with respect to user characteristics

(4) Automatic enhancement or relaxation of security based on the SF and PS parameters

(5) Variability of **W** and **V** keep track of variability in the key-stroke patterns

Performance Summary: The authentication system autonomously learns the keystroke patterns of an individual user and uniquely verifies whether the user is a legitimate one or an impostor. The system also adapts to a user's changing typing patterns. A prototype system, on the average, has yielded ~1.36% Imposter Pass Rate (IPR) and ~2.31% False Authentication Rate (FAR). These results exceed those of existing authentication systems.

3. **V. V. Phoha, A Ray, S. Joshi, S. Vyuruu, S. Phoha, "Hidden Markov Model ("HMM")-Based User Authentication Using Keystroke Dynamics,"** Patent filed through Blakely Sokoloff Taylor & Zafman, LLP.

Background: The multidisciplinary research on anomaly detection and failure pattern discovery in Electromechanical Systems and Networked Robotic systems, conducted under the ARO grant DAAD19-01-06046, has been extended to *Computer User Login Authentication*. The objective is to permit authorized entry and inhibit unauthorized access into computer systems and networks, based on the key stroke typing of individual users. This neural-network-based authentication algorithm makes a suboptimal trade-off between the probabilities of correct authorization and false through classification of the Keystroke-typing patterns as *normal* or *anomalous*.

Key Features: A novel computer user authentication technique using Hidden Markov Model (HMM) has been developed based on key-stroke pattern classification. Authentication of a user is made in two stages: (i) user identification stage, wherein the user's identity is determined by the maximum probability score for the given keystroke pattern, and (ii) user verification stage, wherein the probability score for the given keystroke pattern is examined for a claimed user. Decisions on the authenticity of a user are made based on the results of both the stages and threshold criteria.

Contributions: The significant contributions of the HMM-based authentication algorithm are outlined below:

(1) A novel HMM architecture to build authentication systems

(2) Self adaptation to changes in typing patterns without re-training the network

(3) Robustness of network performance connections with respect to user characteristics

Performance Summary: The authentication system autonomously learns the keystroke patterns of an individual user and uniquely verifies whether the user is a legitimate one or an impostor. The system also adapts to a user's changing typing patterns. A prototype system, on the average, has yielded a false accept rate of ~0.74% and false reject rate of ~8.06%.

APPENDICES

Selected papers published in peer reviewed journals

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Signal Processing 84 (2004) 1115–1130

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

Symbolic dynamic analysis of complex systems for anomaly detection[☆]

Asok Ray*

Mechanical Engineering Department, 137, Reber Building, The Pennsylvania State University, University Park, PA 16802, USA

Received 2 October 2003

Abstract

This paper presents a novel concept of anomaly detection in complex dynamical systems using tools of *Symbolic Dynamics*, *Finite State Automata*, and *Pattern Recognition*, where time-series data of the observed variables on the fast time-scale are analyzed at slow time-scale epochs for early detection of (possible) anomalies. The concept of anomaly detection in dynamical systems is elucidated based on experimental data that have been generated from an active electronic circuit with a slowly varying dissipation parameter.

Keywords: Fault detection; Symbolic dynamics; Pattern recognition; Complex systems

1. Introduction

Anomaly in a dynamical system is defined as a deviation from its nominal behavior and can be associated with parametric or non-parametric changes that may gradually evolve in the system. Early detection of anomalies in complex dynamical systems is essential not only for prevention of cascading catastrophic failures, but also for enhancement of performance and availability [16]. For anomaly detection, it might be necessary to rely on time-series data generated from sensors and other sources of information [1], because accurate and computationally tractable modelling of complex system dynamics is often infeasible solely based on the fundamental principles of physics.

This paper formulates and validates, by laboratory experimentation, a novel concept for detection of slowly evolving anomalies in complex dynamical systems. Often such dynamical systems are either self-excited or can be stimulated with a priori known exogenous inputs to recognize (possible) anomaly patterns from the observed stationary response. Early detection of an anomaly (i.e., small parametric or non-parametric changes) has motivated formulation and validation of the proposed *Symbolic Dynamic* approach to pattern recognition, which is based on the following assumptions:

- The system behavior is stationary at the fast time scale of the process dynamics;

[☆] This work has been supported in part by Army Research Office (ARO) under Grant No. DAAD19-01-1-0646; and NASA Glenn Research Center under Grant No. NNC04GA49G.

* Tel.: +1-814-865-6377; fax: +1-814-863-4848.
E-mail address: axr2@psu.edu (A. Ray).

0165-1684/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2004.03.011

- An observable non-stationary behavior of the dynamical system can be associated with anomaly(ies) evolving at a slow time scale.

The theme of anomaly detection, formulated in this paper, is built upon the concepts of *Symbolic Dynamics* [14,15] *Finite State Automata* [12], and *Pattern Recognition* [9] as a means to qualitatively describe the (fast-time-scale) dynamical behavior in terms of symbol sequences [2,4]. Appropriate phase-space partitioning of the dynamical system yields an alphabet to obtain symbol sequences from time-series data [1,8,13]. Then, tools of computational mechanics [7] are used to identify statistical patterns in these symbolic sequences through construction of a (probabilistic) finite-state machine from each symbol sequence. Transition probability matrices of the finite-state machines, obtained from the symbol sequences, capture the pattern of the system behavior by information compression. For anomaly detection, it suffices that a detectable change in the pattern represents a deviation of the nominal behavior from an anomalous one. The state probability vectors, which are derived from the respective state transition matrices under the nominal and an anomalous condition, yield a vector measure of the anomaly, which provides more information than a scalar measure such as the complexity measure [20].

In contrast to the ε-machine [7,20] that has an a priori unknown structure and yields optimal pattern discovery in the sense of mutual information [5,11], the state machine adopted in this paper has an a priori known structure that can be freely chosen. Although the proposed approach is suboptimal, it provides a common state machine structure where physical significance of each state is invariant under changes in the statistical patterns of symbol sequences. This feature allows unambiguous detection of possible anomalies from symbol sequences at different (slow-time) epochs. The proposed approach is apparently computationally faster than the ε-machine [20], because of significantly fewer number of floating point arithmetic operations. These are the motivating factors for introducing this new anomaly detection concept that is based on a fixed-structure fixed-order Markov chain, called the D-Markov machine in the sequel.

The anomaly detection problem is separated into two parts [21]: (i) *forward problem of Pattern Discovery* to identify variations in the anomalous behavior patterns, compared to those of the nominal behavior; and (ii) *inverse problem of Pattern Recognition* to infer parametric or non-parametric changes based on the learnt patterns and observed stationary response. The inverse problem could be ill-posed or have no unique solution. That is, it may not always be possible to identify a unique anomaly pattern based on the observed behavior of the dynamical system. Nevertheless, the feasible range of parameter variation estimates can be narrowed down from the intersection of the information generated from inverse images of the responses under several stimuli.

It is envisioned that complex dynamical systems will acquire the ability of *self-diagnostics* through usage of the proposed anomaly detection technique that is analogous to the diagnostic procedure employed in medical practice in the following sense. Similar to the notion of injecting *medication* or *inoculation* on a nominally healthy patient, a dynamical system would be excited with known stimuli (chosen in the forward problem) in the idle cycles for self diagnosis and health monitoring. The inferred information on health status can then be used for the purpose of self-healing control or life-extending control [25]. This paper focuses on the forward problem and demonstrates the efficacy of anomaly detection based on experimental data generated from an active electronic circuit with a slowly varying dissipation parameter.

The paper is organized in seven sections and two appendices. Section 2 briefly introduces the notion of nonlinear time-series analysis. Section 3 provides a brief overview of symbolic dynamics and encoding of time series data. Section 4 presents two ensemble approaches for statistical pattern representation. It also presents information extraction based on the ε-machine [7] and the D-Markov machine, as well as their comparison from different perspectives. Section 5 presents the notion of anomaly measure to quantify the changing patterns of anomalous behavior of the dynamical system form the information-theoretic perspectives, followed by an outline of the anomaly detection procedure. Section 6 presents experimental results on a nonlinear active electronic circuit to demonstrate efficacy of the proposed anomaly detection technique. Section 7 summarizes and concludes the paper with recommendations for future research. Appendix A explains the physical significance of information-theoretic quantities used in the Section 4.1 and Section 5. Appendix B introduces the concept of shift spaces, which is used to delineate the differences between the ε-machine [7] and the D-Markov machine in Section 4.4.

2. Nonlinear time-series analysis

This section presents nonlinear time-series analysis (NTSA) that is needed to extract relevant physical information on the dynamical system from the observed data. NTSA techniques are usually executed in the following steps [1]:

1. *Signal separation*: The (deterministic) time-dependent signal $\{y(n) : n \in \mathbb{N}\}$, where $\mathbb{N}$ is the set of positive integers, is separated from noise, using time-frequency and other types of analysis.

2. *Phase space reconstruction*: Based on the Takens Embedding theorem [22], time lagged or delayed variables are used to construct the state vector $\mathbf{x}(n)$ in a phase space of dimension d_E (which is diffeomorphically equivalent to the attractor of the original dynamical system) as follows:

$$\mathbf{x}(n) = [y(n), y(n+T), \ldots, y(n+(d_E-1)T)], \quad (1)$$

where the time lag T is determined using *mutual information*; and one of the ways to determine d_E is the *false nearest neighbors test* [1].

3. *Signal classification*: Signal classification and system identification in nonlinear chaotic systems require a set of invariants for each subsystem of interest followed by comparison of observations with those in the library of invariants. The invariants are properties of the attractor and could be independent of any particular trajectory. These invariants can be divided into two classes: *fractal dimensions* and *Lyapunov exponents*. Fractal dimensions characterize geometrical complexity of dynamics (e.g., spatial distribution of points along a system orbit); and Lyapunov exponents describe the dynamical complexity (e.g., stretching and folding of an orbit in the phase space) [18].

4. *Modeling and prediction*: This step involves determination of the parameters of the assumed model of the dynamics, which is consistent with the invariant classifiers (e.g., Lyapunov exponents, and fractal dimensions).

The first three steps show how chaotic systems may be separated from stochastic ones and, at the same time, provide estimates of the degrees of freedom and the complexity of the underlying dynamical system. Based on this information, Step 4 formulates a state-space model that can be used for prediction of anomalies and incipient failures. The functional form often used in this step, includes orthogonal polynomials and radial basis functions. This paper has adopted an alternative class of discrete models inspired from *Automata Theory*, which is built upon the principles of *Symbolic Dynamics* as described in the following section.

3. Symbolic dynamics and encoding

This section introduces the concept of *Symbolic Dynamics* and its usage for encoding nonlinear system dynamics from observed time-series data. Let a continuously varying physical process be modelled as a finite-dimensional dynamical system in the setting of an initial value problem:

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = f(\mathbf{x}(t), \theta(t_s)); \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (2)$$

where $t \in [0, \infty)$ denotes the (fast-scale) time; $\mathbf{x} \in \mathbb{R}^n$ is the state vector in the phase space; and $\boldsymbol{\theta} \in \mathbb{R}^\ell$ is the (possibly anomalous) parameter vector varying in (slow-scale) time t_s. Sole usage of the model in Eq. (2) may not always be feasible due to unknown parametric and non-parametric uncertainties and noise. A convenient way of learning the dynamical behavior is to rely on the additional information provided by (sensor-based) time-series data [1,4].

A tool for behavior description of nonlinear dynamical systems is based on the concept of formal languages for transitions from smooth dynamics to a discrete symbolic description [2]. The phase space of the dynamical system in Eq. (2) is partitioned into a finite number of cells, so as to obtain a coordinate grid of the space. A compact (i.e., closed and bounded) region $\Omega \in \mathbb{R}^n$, within which the (stationary) motion under the specific exogenous stimulus is circumscribed, is identified. Encoding of Ω is accomplished by introducing a partition $\mathbb{B} \equiv \{B_0, \ldots, B_{m-1}\}$ consisting of m mutually exclusive (i.e., $B_j \cap B_k = \emptyset \ \forall j \neq k$), and exhaustive (i.e., $\bigcup_{j=0}^{m-1} B_j = \Omega$) cells. The dynamical system describes an orbit by the time-series data as:

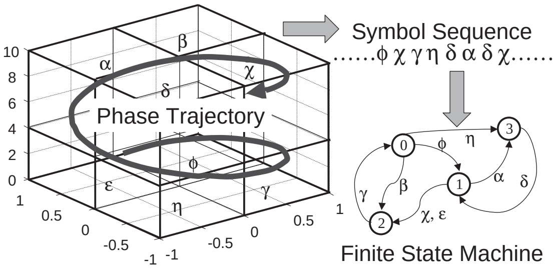


Fig. 1. Continuous dynamics to symbolic dynamics.

$\mathbb{O} \equiv \{x_0, x_1 \cdots, x_k \cdots\}, x_i \in \Omega$, which passes through or touches the cells of the partition $\mathbb{B}$.

Let us denote the cell visited by the trajectory at a time instant as a random variable S that takes a symbol value $s \in \mathscr{A}$. The set $\mathscr{A}$ of m distinct symbols that label the partition elements is called the *symbol alphabet*. Each initial state $x_0 \in \Omega$ generates a sequence of symbols defined by a mapping from the phase space into the symbol space as:

$$x_0 \longrightarrow s_{i0} s_{i1} s_{i2} \cdots s_{ik} \cdots \ . \tag{3}$$

The mapping in Eq. (3) is called *Symbolic Dynamics* as it attributes a legal (i.e., physically admissible) symbol sequence to the system dynamics starting from an initial state. (Note: A symbol alphabet $\mathscr{A}$ is called a generating partition of the phase space Ω if every legal symbol sequence uniquely determines a specific initial condition x_0, i.e., every symbolic orbit uniquely identifies one continuous space orbit.) Fig. 1 pictorially elucidates the concepts of partitioning a finite region of the phase space and mapping from the partitioned space into the symbol alphabet. This represents a spatial and temporal discretization of the system dynamics defined by the trajectories. Fig. 1 also shows conversion of the symbol sequence into a finite-state machine as explained in later sections.

Symbolic dynamics can be viewed as coarse graining of the phase space, which is subjected to (possible) loss of information resulting from granular imprecision of partitioning boxes, measurement noise and errors, and sensitivity to initial conditions. However, the essential robust features (e.g., periodicity and chaotic behavior of an orbit) are expected to be preserved in the symbol sequences through an appropriate partitioning of the phase space [2]. Although the theory of phase-space partitioning is well developed for one-dimensional mappings, very few results are known for two- and higher-dimensional systems [4].

4. Pattern identification

Given the intricacy of phase trajectories in complex dynamical systems, the challenge is to identify their *patterns* in an appropriate category by using one of the following two alternative approaches:

- The single-item approach, which relies on Kolmogorov Chiatin (KC) complexity, also known as algorithmic complexity [5], for exact pattern regeneration;
- The ensemble approach, which regards the pattern as one of many possible experimental outcomes, for estimated pattern regeneration.

While the single-item approach is common in coding theory and computer science, the ensemble approach has been adopted in this paper due to its physical and statistical relevance. As some of the legal symbol sequences may occur more frequently than others, a probability is attributed to each observed sequence. The collection of all legal symbol sequences $S_{-M} \cdots S_{-2} S_{-1} S_0 S_1 \cdots S_N$, $N, M = 0, 1, 2 \ldots$, defines a stochastic process that is a symbolic probabilistic description of the continuous system dynamics.

Let us symbolically denote a discrete-time, discrete-valued stochastic process as

$$\mathbb{S} \equiv \cdots, S_{-2} S_{-1} S_0 S_1 S_2 \cdots, \tag{4}$$

where each random variable S_i takes exactly one value in the (finite) alphabet $\mathscr{A}$ of m symbols (see Section 3). The symbolic stochastic process $\mathbb{S}$ is dependent on the specific partitioning of the phase space and is non-Markovian, in general. Even if a partitioning that makes the stochastic process a Markov chain exists, identification of such a partitioning is not always feasible because the individual cells may have fractal boundaries instead of being simple geometrical objects. In essence, there is a trade-off between selecting a simple partitioning leading to a complicated stochastic process, and a complicated partitioning leading to a simple stochastic process. Recent literature has reported a comprehensive numerical procedure for construction phase-space partitions from the time-series data [13]. Having defined a partition of the phase space, the time-series data is converted to a symbol sequence that, in turn, is used for construction of a finite-state machine using the tools of Computational Mechanics [7] as illustrated in Fig. 1.

This paper considers two alternative techniques of finite-state machine construction from a given symbol sequence $\mathscr{S}$: (i) the ε-machine formulation [20]; and (ii) a new concept based on Dth order Markov chains, called the D-Markov machine, for identifying patterns based on time series analysis of the observed data. Both techniques rely on information-theoretic principles (see Appendix A) and are based on computational mechanics [7].

4.1. The ε-machine

Like statistical mechanics [10,4], computational mechanics is concerned with dynamical systems consisting of many partially correlated components. Whereas Statistical Mechanics deals with the local space–time behavior and interactions of the system elements, computational mechanics relies on the joint probability distribution of the phase-space trajectories of a dynamical system. The ε-machine construction [7,20] makes use of the joint probability distribution to infer the information processing being performed by the dynamical system. This is developed using the statistical mechanics of orbit ensembles, rather than focusing on the computational complexity of individual orbits.

Let the symbolic representation of a discrete-time, discrete-valued stochastic process be denoted by: $\mathbb{S} \equiv \cdots S_{-2}S_{-1}S_0S_1S_2\cdots$ as defined earlier in Section 4. At any instant t, this sequence of random variables can be split into a sequence $\overleftarrow{S}_t$ of the past and a sequence $\overrightarrow{S}_t$ of the future. Assuming conditional stationarity of the symbolic process $\mathbb{S}$ (i.e., $P[\overleftarrow{S}_t|\overrightarrow{S}_t = \overrightarrow{s}]$ being independent of t), the subscript t can be dropped to denote the past and future sequences as $\overleftarrow{S}$ and $\overrightarrow{S}$, respectively. A symbol string, made of the first L symbols of $\overrightarrow{S}$, is denoted by $\overrightarrow{S}^L$. Similarly, a symbol string, made of the last L symbols of $\overleftarrow{S}$, is denoted by $\overleftarrow{S}^L$.

Prediction of the future $\overrightarrow{S}$ necessitates determination of its probability conditioned on the past $\overleftarrow{S}$, which requires existence of a function ε mapping histories $\overleftarrow{s}$ to predictions $P(\overrightarrow{S}|\overleftarrow{s})$. In essence, a prediction imposes a partition on the set $\overleftarrow{\mathbf{S}}$ of all histories. The cells of this partition contain histories for which the same prediction is made and are called the *effective states* of the process under the given predictor. The set of effective states is denoted by $\mathbf{R}$; a random variable for an effective state is denoted by $\mathscr{R}$ and its realization by ρ.

The objective of ε-machine construction is to find a predictor that is an optimal partition of the set $\overleftarrow{\mathbf{S}}$ of histories, which requires invoking two criteria in the theory of Computational Mechanics [6]:

1. *Optimal Prediction*: For any partition of histories or effective states $\mathscr{R}$, the conditional entropy $H[\overrightarrow{S}^L|\mathscr{R}] \geqslant H[\overrightarrow{S}^L|\overleftarrow{S}]$, $\forall L \in \mathbb{N}$, $\forall \overleftarrow{S} \in \overleftarrow{\mathbf{S}}$, is equivalent to remembering the whole past. Effective states $\mathscr{R}$ are called *prescient* if the equality is attained $\forall L \in \mathbb{N}$. Therefore, optimal prediction needs the effective states to be prescient.

2. *Principle of Occam Razor*: The prescient states with the least complexity are selected, where complexity is defined as the measured Shannon information of the effective states:

$$H[\mathscr{R}] = -\sum_{\rho\in\mathbf{R}} P(\mathscr{R}=\rho)\log P(\mathscr{R}=\rho). \tag{5}$$

Eq. (5) measures the amount of past information needed for future prediction and is known as *Statistical Complexity* denoted by $C_\mu(\mathscr{R})$ (see Appendix A).

For each symbolic process $\mathbb{S}$, there is a unique set of prescient states known as *causal states* that minimize the statistical complexity $C_\mu(\mathscr{R})$.

Definition 4.1 (Shalizi et al. [20]). Let $\mathbb{S}$ be a (conditionally) stationary symbolic process and $\overleftarrow{\mathbf{S}}$ be the set of histories. Let a mapping $\varepsilon : \overleftarrow{\mathbf{S}} \to \Upsilon(\overrightarrow{\mathbf{S}})$ from the set $\overleftarrow{\mathbf{S}}$ of histories into a collection $\Upsilon(\overrightarrow{\mathbf{S}})$ of measurable subsets of $\overleftarrow{\mathbf{S}}$ be defined as:

$$\forall \Gamma \in \Upsilon(\overrightarrow{\mathbf{S}}), \quad \varepsilon(\overleftarrow{s}) \equiv \{\overleftarrow{s}' \in \overleftarrow{\boldsymbol{S}} \text{ such that}$$

$$P(\overrightarrow{S} \in \Gamma|\overleftarrow{S} = \overleftarrow{s}) = P(\overrightarrow{S} \in \Gamma|\overleftarrow{S} = \overleftarrow{s}')\}. \tag{6}$$

Then, the members of the range of the function ε are called the causal states of the symbolic process $\mathbb{S}$. The ith causal state is denoted by q_i and the set of all causal states by $\mathbf{Q} \subseteq \Upsilon(\overrightarrow{\mathbf{S}})$. The random variable corresponding to a causal state is denoted by $\mathscr{Q}$ and its realization by q.

Given an initial causal state and the next symbol from the symbolic process, only successor causal states are possible. This is represented by the legal transitions among the causal states, and the probabilities of these transitions. Specifically, the probability of transition from state q_i to state q_j on a single symbol s is expressed as:

$$T_{ij}^{(s)} = P(\overrightarrow{S}^1 = s, \mathscr{Q}' = q_j \mid \mathscr{Q} = q_i)\ \forall q_i, q_j \in \mathbf{Q}, \tag{7}$$

$$\sum_{s \in \mathscr{A}} \sum_{q_j \in \mathbf{Q}} T_{ij}^{(s)} = 1. \tag{8}$$

The combination of causal states and transitions is called the *ε-machine* (also known as *the causal state model* [20]) of a given symbolic process. Thus, the ε-machine represents the way in which the symbolic process stores and transforms information. It also provides a description of the pattern or regularities in the process, in the sense that the pattern is an algebraic structure determined by the causal states and their transitions. The set of labelled transition probabilities can be used to obtain a stochastic matrix [3] given by: $\mathscr{T} = \sum_{s \in \mathscr{A}} \mathscr{T}^s$ where the square matrix $\mathscr{T}^s$ is defined as: $\mathscr{T}^s = [T_{ij}^s]\ \forall s \in \mathscr{A}$. Denoting $\mathbf{p}$ as the left eigenvector of $\mathscr{T}$, corresponding to the eigenvalue $\lambda = 1$, the probability of being in a particular causal state can be obtained by normalizing $\|\mathbf{p}\|_{\ell_1} = 1$. A procedure for construction of the ε-machine is outlined below.

The original ε-machine construction algorithm is the subtree-merging algorithm as introduced in [7,6]. The default assumption of this technique was employed by Surana et al. [21] for anomaly detection. This approach has several shortcomings, such as lack of a systematic procedure for choosing the algorithm parameters, may return non-deterministic causal states, and also suffers from slow convergence rates. Recently, Shalizi et al. [20] have developed a code known as Causal State Splitting Reconstruction (CSSR) that is based on state splitting instead of state merging as was done in the earlier algorithm of subtree-merging [7]. The CSSR algorithm starts with a simple model for the symbolic process and elaborates the model components only when statistically justified. Initially, the algorithm assumes the process to be independent and identically distributed (iid) that can be represented by a single causal state and hence zero statistical complexity and high entropy rate. At this stage, CSSR uses statistical tests to determine when it must add states to the model, which increases the estimated complexity, while lowering the entropy rate h_μ (see Appendix A). A key and distinguishing feature of the CSSR code is that it maintains homogeneity of the causal states and deterministic state-to-state transitions as the model grows. Complexity of the CSSR algorithm is: $\mathrm{O}(m^{L_{\max}}) + \mathrm{O}(m^{2L_{\max}+1}) + \mathrm{O}(N)$, where m is the size of the alphabet $\mathscr{A}$; N is the data size and $L_{\max}$ is the length of the longest history to be considered. Details are given in [20].

4.2. The suboptimal D-markov machine

This section presents a new alternative approach for representing the pattern in a symbolic process, which is motivated from the perspective of anomaly detection. The core assumption here is that the symbolic process can be represented to a desired level of accuracy as a Dth order Markov chain, by appropriately choosing $D \in \mathbb{N}$.

Definition 4.2. A stochastic symbolic stationary process $\mathbb{S} \equiv \cdots S_{-2}S_{-1}S_0S_1S_2\cdots$ is called Dth order Markov process if the probability of the next symbol depends only on the previous (at most) D symbols, i.e. the following condition holds:

$$P(S_i|S_{i-1}S_{i-2}\cdots S_{i-D}\cdots) = P(S_i|S_{i-1}\cdots S_{i-D}) \tag{9}$$

Alternatively, symbol strings $\overleftarrow{S}, \overleftarrow{S}' \in \overleftarrow{\mathbf{S}}$ become indistinguishable whenever the respective substrings $\overleftarrow{S}^D$ and $\overleftarrow{S}'^D$, made of the most recent D symbols, are identical.

Definition (4.2) can be interpreted as follows: $\forall\ \overleftarrow{S}, \overleftarrow{S}' \in \overleftarrow{\mathbf{S}}$ such that $|\overleftarrow{S}| \geqslant D$ and $|\overleftarrow{S}'| \geqslant D$, $(\overleftarrow{S}' \in \varepsilon(\overleftarrow{S})$ and $\overleftarrow{S} \in \varepsilon(\overleftarrow{S}'))$ iff $\overleftarrow{S}^D = \overleftarrow{S}'^D$. Thus, a set $\{\overleftarrow{S}^L : L \geqslant D\}$ of symbol stings can be partitioned into a maximum of $|\mathscr{A}|^D$ equivalence classes where $\mathscr{A}$ is the symbol alphabet, under the equivalence relation defined in Eq. (6). Each symbol string in $\{\overleftarrow{S}^L : L \geqslant D\}$ either belongs to one of the $|\mathscr{A}|^D$ equivalence classes or has a distinct equivalence class. All such symbol strings belonging to the distinct equivalence class form transient states, and would not be of concern to anomaly detection for a (fast-time-scale) stationary condition under (slowly changing) anomalies. Given $D\in\mathbb{N}$ and a symbol string $\overleftarrow{s}$ with $|\overleftarrow{s}|=D$, the *effective* state $q(D,\overleftarrow{s})$ is the equivalence class of symbol strings as defined below:

$$q(D,\overleftarrow{s})=\{\overleftarrow{S}\in\overleftarrow{\mathbf{S}}\colon\ \overleftarrow{S}^D=\overleftarrow{s}\}\tag{10}$$

and the set $\mathbf{Q}(D)$ of *effective* states of the symbolic process is the collection of all such equivalence classes. That is,

$$\mathbf{Q}(D)=\{q(D,\overleftarrow{s})\colon\ \overleftarrow{s}\in\overleftarrow{\mathbf{S}}^D\}\tag{11}$$

and hence $|\mathbf{Q}(D)|=|\mathscr{A}|^D$. A random variable for a state in the above set $\mathbf{Q}$ of states is denoted by $\mathscr{Q}$ and the jth state as q_j. The probability of transitions from state q_j to state q_k is defined as:

$$\pi_{jk}=P(s\in\overrightarrow{S}^1\,|\,q_j\in\mathbf{Q},\ (s,q_j)\to q_k);$$

$$\sum_k\pi_{jk}=1;\tag{12}$$

Given an initial state and the next symbol from the original process, only certain successor states are accessible. This is represented as the allowed state transitions resulting from a single symbol. Note that $\pi_{ij}=0$ if $s_2s_3\cdots s_D\neq s'_1\cdots s'_{D-1}$ whenever $q_i\equiv s_1s_2\cdots s_D$ and $q_j\equiv s'_1s'_2\cdots s'_D$. Thus, for a D-Markov machine, the stochastic matrix $\Pi\equiv[\pi_{ij}]$ becomes a branded matrix with at most $|\mathscr{A}|^{D+1}$ nonzero entries.

The construction of a D-Markov machine is fairly straightforward. Given $D\in\mathbb{N}$, the states are as defined in Eqs. (10) and (11). On a given symbol sequence $\mathscr{S}$, a window of length $(D+1)$ is slided by keeping a count of occurrences of sequences $s_{i_1}\cdots s_{i_D}s_{i_{D+1}}$ and $s_{i_1}\cdots s_{i_D}$ which are, respectively, denoted by $N(s_{i_1}\cdots s_{i_D}s_{i_{D+1}})$ and $N(s_{i_1}\cdots s_{i_D})$. Note that if $N(s_{i_1}\cdots s_{i_D})=0$, then the state $q\equiv s_{i_1}\cdots s_{i_D}\in\mathbf{Q}$ has zero probability of occurrence. For $N(s_{i_1}\cdots s_{i_D})\neq 0$, the transitions probabilities are then obtained by these frequency counts as follows:

$$\pi_{jk}=\frac{P(s_{i_1}\cdots s_{i_D}s)}{P(s_{i_1}\cdots s_{i_D})}\approx\frac{N(s_{i_1}\cdots s_{i_D}s)}{N(s_{i_1}\cdots s_{i_D})},\tag{13}$$

where the corresponding states are denoted by: $q_j\equiv s_{i_1}s_{i_2}\cdots s_{i_D}$ and $q_k\equiv s_{i_2}\cdots s_{i_D}s$.

As an example, Fig. 2 shows the finite-state machine and the associated state transition matrix for a D-Markov machine, where the alphabet $\mathscr{A}=\{0,1\}$, i.e., alphabet size $|\mathscr{A}|=2$; and the states are chosen as words of length $D=2$ from a symbol sequence $\mathscr{S}$. Consequently, the total number of states is $|\mathscr{A}|^D=4$, which is the number of permutations of the alphabet symbols within a word of length D; and the set of states $\mathbf{Q}=\{00,01,10,11\}$. The state transition matrix on the right half of Fig. 2 denotes the probability $\pi_{ij}=p_{ij}$ of occurrence of the symbol $0\in\mathscr{A}$ at the state $q\equiv ij$, where $i,j\in\mathscr{A}$. The states are joined by edges labelled by a symbol in the alphabet. The state machine moves from one state to another upon occurrence of an event as a new symbol in the symbol sequence is received and the resulting transition matrix has at most $|\mathscr{A}|^{D+1}=8$ non-zero entries. The machine language is complete in the sense that there are different outgoing edges marked by different symbols; however, it is possible that some of these arcs may have zero probability.

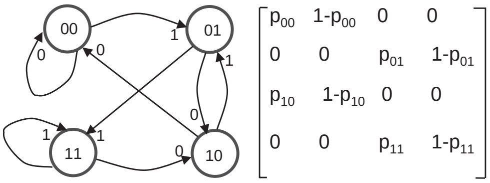


Fig. 2. State machine with $D=2$, and $|\mathscr{A}|=2$.

4.3. Statistical mechanical concept of D-Markov machine

This section outlines an analogy between the structural features of the D-Markov machine and those of spin models in Statistical Mechanics. The main idea is derived from the doctoral dissertation of Feldman [10] who has demonstrated how measures of patterns from Information Theory and Computational Mechanics are captured in the construction of ε-Machines. In general, the effects of an anomaly are reflected in the respective state transition matrices. Thus, the structure of the finite-state machine is fixed for a given alphabet size $|\mathscr{A}|$ and window length D. Furthermore, the number of edges is also finite because of the finite alphabet size. The elements of the state transition matrix (that is a stochastic matrix [3]) are identified from the symbol sequence.

For $|\mathscr{A}| = 2$ and $D = 2$, the finite-state machine construction is (to some extent) analogous to the one-dimensional Ising model of spin-1/2 systems with nearest neighbor interactions, where the z-component of each spin takes on one of the two possible values $s = +1$ or $s = -1$ [10,19]. For $|\mathscr{A}| \geqslant 3$, the machine would be analogous to one-dimensional Potts model, where each spin is directed in the z-direction with $|\mathscr{A}|$ different discrete values s_k: $k \in 1, 2, \ldots, |\mathscr{A}|$; for a $j/2$-spin model, the alphabet size $|\mathscr{A}| = j + 1$ [4]. For $D \geqslant 2$, the spin interactions extend up to the $(D-1)$th neighbor.

4.4. Comparison of *D*-Markov machine and ε-machine

An ε-machine seeks to find the patterns in the time series data in the form of a finite-state machine, whose states are chosen for optimal prediction of the symbolic process; and a finite-state automation can be used as a pattern for prediction [20]. An alternative notion of the pattern is one which can be used to compress the given observation. The first notion of the pattern subsumes the second, because the capability of optimal prediction necessarily leads to the compression as seen in the construction of states by lumping histories together. However, the converse is not true in general. For the purpose of anomaly detection, the second notion of pattern is sufficient because the goal is to represent and detect the deviation of an anomalous behavior from the nominal behavior. This has been the motivating factor for proposing an alternative technique, based on the fixed structure D-Markov machine. It is possible to detect the evolving anomaly, if any, as a change in the probability distribution over the states.

Another distinction between the D-Markov machine and ε-machine can be seen in terms of *finite-type shifts* and *sofic shifts* [15] (see Appendix B). Basic distinction between finite-type shifts and sofic shifts can be characterized in terms of the *memory*: while a finite-type shift has finite-*length* memory, a sofic shift uses finite *amount* of memory in representing the patterns. Hence, finite-type shifts are strictly proper subsets of sofic shifts. While, any finite-type shift has a representation as a graph, sofic shifts can be represented as a *labelled graph*. As a result, the finite-type shift can be considered as an "extreme version" of a D-Markov chain (for an appropriate D) and sofic shifts as an "extreme version" of a Hidden Markov process [24], respectively. The shifts have been referred to as "extreme" in the sense that they specify only a set of allowed sequences of symbols (i.e., symbol sequences that are actually possible, but not the probabilities of these sequences). Note that a Hidden Markov model consists of an internal D-order Markov process that is observed only by a function of its internal-state sequence. This is analogous to sofic shifts that are obtained by a labelling function on the edge of a graph, which otherwise denotes a finite-type shift. Thus, in these terms, an ε-machine infers the Hidden Markov Model (sofic shift) for the observed process. In contrast, the D-Markov Model proposed in this paper infers a (finite-type shift) approximation of the (sofic shift) ε-machine.

5. Anomaly measure and detection

The machines described in Sections 4.1 and 4.2 recognize patterns in the behavior of a dynamical system that undergoes anomalous behavior. In order to quantify changes in the patterns that are representations of evolving anomalies, we induce an *anomaly measure* on these machines, denoted by $\mathscr{M}$. The anomaly measure $\mathscr{M}$ can be constructed based on the following information-theoretic quantities: entropy rate, excess entropy, and complexity measure of a symbol string $\mathscr{S}$ (see Appendix A).

- The entropy rate $h_\mu(\mathscr{S})$ quantifies the intrinsic randomness in the observed dynamical process.
- The excess entropy $\mathbf{E}(\mathscr{S})$ quantifies the memory in the observed process.
- The statistical complexity $\mathscr{C}_\mu(\mathscr{S})$ of the state machine captures the average memory requirements for modelling the complex behavior of a process.

Given two symbol strings $\mathscr{S}$ and $\mathscr{S}_0$, it is possible to obtain a measure of anomaly by adopting any one of the following three alternatives:

$$\mathscr{M}(\mathscr{S}, \mathscr{S}_0)) = \begin{cases} |h_\mu(\mathscr{S}) - h_\mu(\mathscr{S}_0)|, \text{ or} \\ |\mathbf{E}(\mathscr{S}) - \mathbf{E}(\mathscr{S}_0)|, \text{ or} \\ |C_\mu(\mathscr{S}) - C_\mu(\mathscr{S}_0)|. \end{cases}$$

Note that each of the anomaly measures, defined above, is a *pseudo metric* [17]. For example, let us consider two periodic processes with unequal periods, represented by $\mathscr{S}$ and $\mathscr{S}_0$. For both processes, $h_\mu = 0$, so that $\mathscr{M}(\mathscr{S}, \mathscr{S}_0) = 0$ for the first of the above three options, even if $\mathscr{S} \neq \mathscr{S}_0$.

The above measures are obtained through scalar-valued functions defined on a state machine and do not exploit the rich algebraic structure represented in the state machine. For example, the connection matrix $\mathscr{T}$ associated with the ε-machine (see Section 4.1), can be treated as a vector representation of any possible anomalies in the dynamical system. The induced 2-*norm* of the difference between the $\mathscr{T}$-matrices for the two state machines can then be used as a measure of anomaly, i.e., $\mathscr{M}(\mathscr{S}, \mathscr{S}_0) = \|\mathscr{T} - \mathscr{T}_0\|_2$. Such a measure, used in [21], was found to be effective. However, there is some subtlety in using this measure on ε-machines, because ε-machines do not guarantee that the machines formulated from the symbol sequences $\mathscr{S}$ and $\mathscr{S}_0$ have the same number of states; and these states do not necessarily have similar physical significance. In general, $\mathscr{T}$ and $\mathscr{T}_0$ may have different dimensions and different physical significance. However, by encoding the causal states, $\mathscr{T}$ could be embedded in a larger matrix, and an induced norm of the difference between $\mathscr{T}$ matrices for these two machines can be defined. Alternatively, a (vector) measure of anomaly can be derived directly from the stochastic matrix $\mathscr{T}$ as the left eigenvector **p** corresponding to the unit eigenvalue of $\mathscr{T}$, which is the state probability vector under a stationary condition.

This paper has adopted the *D*-Markov machine approach, described in the Section 4.2 to build the state machines. Since *D*-Markov machines have a fixed state structure, the state probability vector **p** associated with the state machine have been used for a vector representation of anomalies, leading to the anomaly measure $\mathscr{M}(\mathscr{S}, \mathscr{S}_0))$ as a distance function between the respective probability vectors **p** and $\mathbf{p}_0$ (that are of identical dimensions), or any other appropriate functional.

5.1. Anomaly detection procedure

Having discussed various tools and techniques, this section outlines the steps of the *forward problem* and the *inverse problem* described in Section 1. Following are the steps for the forward problem:

(F1) Selection of an appropriate set of input stimuli.
(F2) Signal–noise separation, time interval selection, and phase-space construction.
(F3) Choice of a phase space partitioning to generate symbol alphabet and symbol sequences.
(F4) State Machine construction using generated symbol sequence(s) and determining the connection matrix.
(F5) Selection of an appropriate metric for the anomaly measure $\mathscr{M}$.
(F6) Formulation and calibration of a (possibly non-parametric) relation between the computed anomaly measure and known physical anomaly under which the time-series data were collected at different (slow-time) epochs.

Following are the steps for the inverse problem:

(I1) Excitation with known input stimuli selected in the forward problem.
(I2) Generation of the stationary behavior as time-series data for each input stimulus at different (slow-time) epochs.
(I3) Embedding the time-series data in the phase space determined for the corresponding input stimuli in Step F2 of the forward problem.
(I4) Generation of the symbol sequence using the same phase-space partition as in Step F3 of the forward problem.
(I5) State Machine construction using the symbol sequence and determining the anomaly measure.
(I6) Detection and identification of an anomaly, if any, based on the computed anomaly measure and the relation derived in Step F6 of the forward problem.

6. Application to an active electronic circuit

This section illustrates an application of the *D*-Markov machine concept for anomaly detection on an experimental apparatus that consists of an active electronic circuit. The apparatus implements a second-order non-autonomous, forced Duffing equation in real time [23]. The governing equation with a cubic nonlinearity in one of the state variables is

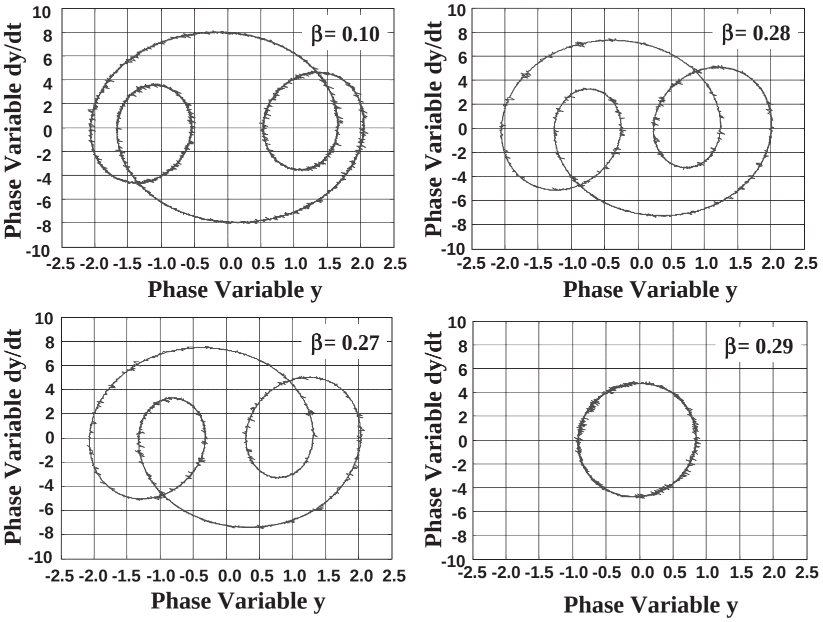


Fig. 3. Phase plots for electronic circuit experiment.

given below:

$$\frac{d^2x(t)}{dt^2} + \beta(t_s)\frac{dx(t)}{dt} + x(t) + x^3(t) = A\cos\omega t. \quad (14)$$

The dissipation parameter $\beta(t_s)$, realized in the form of a resistance in the circuit, is made to vary in the slow time scale t_s and is treated as a constant in the fast time scale t at which the dynamical system is excited. The goal is to detect, at an early stage, changes in $\beta(t_s)$ that are associated with the anomaly.

In the forward problem, the first task is the selection of appropriate input stimuli. For illustration purposes, we have used the stimulus with amplitude $A = 22.0$ and frequency $\omega = 5.0$ in this paper. Changes in the stationary behavior of the electronic circuit take place starting from $\beta \approx 0.10$ with significant changes occurring in the narrow range of $0.28 < \beta < 0.29$. The stationary behavior of the system response for this input stimulus is obtained for several values of β in the range of 0.10–0.35.

The four plates in Fig. 3 exhibit four phase plots for the values of the parameter β at 0.10, 0.27, 0.28, and 0.29, respectively, relating the phase variable of electrical charge that is proportional to the voltage across one of the capacitors in the electronic circuit, and its time derivative (i.e., the instantaneous current). While a small difference between the plots for $\beta = 0.10$ and 0.27 is observed, there is no clearly visible difference between the plots for $\beta = 0.27$ and 0.28 in Fig. 3. However, the phase plots for $\beta = 0.28$ and 0.29 display a very large difference indicating period doubling possibly due to onset of bifurcation. Fig. 4 displays time responses of the stationary behavior of the phase variable for different values of the parameter β corresponding to the phase plots in Fig. 3. The plots in Fig. 4 are phase-aligned for better visibility. (Note that the proposed anomaly detection method does not require phase alignment; equivalently, the finite-state machine Fig. 2 can be started from any arbitrary state corresponding to no specific initial condition.) While the time responses for $\beta = 0.27$ and 0.28 are

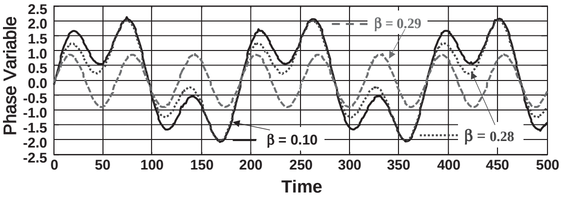


Fig. 4. Time plots for electronic circuit experiment.

indistinguishable, there is a small difference between those for $\beta = 0.27$ and 0.10. Similar to the phase plots in Fig. 3, the time responses for $\beta = 0.28$ and 0.29 display existence of period doubling due to a possible bifurcation.

Additional exogenous stimuli have been identified, which also lead to significant changes in the stationary behavior of the electronic system dynamics for other ranges of β. For example, with the same amplitude $A = 22.0$, stimuli at the excitation frequencies of $\omega = 2.0$ and $\omega = 1.67$ (not shown in Figs. 3 and 4) detect small changes in the ranges of $0.18 < \beta < 0.20$ and $0.11 < \beta < 0.12$, respectively [21]. These observations reveal that exogenous stimuli at different excitation frequencies can be effectively used for detection of small changes in β over an operating range (e.g., $0.10 < \beta < 0.35$).

Having obtained the phase plots from the time-series data, the next step is to find a partition of the phase space for symbol sequence generation. This is a difficult task especially if the time-series data is noise-contaminated. Several methods of phase-space partitioning have been suggested in literature (for example, [1,8,13]). Apparently, there exist no well-established procedure for phase-space partitioning of complex dynamical systems; this is a subject of active research. In this paper, we have introduced a new concept of symbol sequence generation, which uses wavelet transform to convert the time-series data to time-frequency data for generating the symbol sequence. The graphs of wavelet coefficients versus scale at selected time shifts are stacked starting with the smallest value of scale and ending with its largest value and then back from the largest value to the smallest value of the scale at the next instant of time shift. The resulting *scale series* data in the wavelet space is analogous to the time-series data in the phase space. Then, the wavelet space is partitioned into segments of coefficients on the ordinate separated by horizontal lines. The number of segments in a partition is equal to the size of the alphabet and each partition is associated with a symbol in the alphabet. For a given stimulus, partitioning of the wavelet space must remain invariant at all epochs of the slow time scale. Nevertheless, for different stimuli, the partitioning could be chosen differently. (The concept of proposed wavelet-space partitioning would require significant theoretical research before its acceptance for application to a general class of dynamical systems for anomaly detection; and its efficacy needs to be compared with that of existing phase-space partitioning methods such as false nearest neighbor partitioning [13].)

The procedure, described in the subsection IV-B constructs a D-Markov machine and obtains the connection matrix $\mathscr{T}$ and the state vector $\mathbf{p}$ from the symbol sequence corresponding to each β. For this analysis, the wave space generated from each data set has been partitioned into eight (8) segments, which makes the alphabet size $|\mathscr{A}| = 8$ to generate symbol sequences from the *scale series* data. At each value of β, the generated symbol sequence has been used to construct several D-Markov Machines starting with $D = 1$ and higher integers. It is observed that, the dominant probabilities of the state vector (albeit having different dimensions) for different values of D are virtually similar. Therefore, a fixed-structure D-Markov Machine with alphabet size $|\mathscr{A}| = 8$ and depth $D = 1$, which yields the number of states $|\mathscr{A}|^D = 8$, is chosen to generate state probability ($\mathbf{p}$) vectors for the symbol sequences.

The electronic circuit system is assumed to be at the nominal condition for the dissipation parameter $\beta = 0.10$, which is selected as the reference point for calculating the anomaly measure. The anomaly measure $\mathscr{M}$ is computed based on two different computation methods as discussed in Section 5. Fig. 5 exhibits three plots of the normalized anomaly measure $\mathscr{M}$ versus the dissipation parameter β, where $\mathscr{M}$ is computed based on different metrics. In each

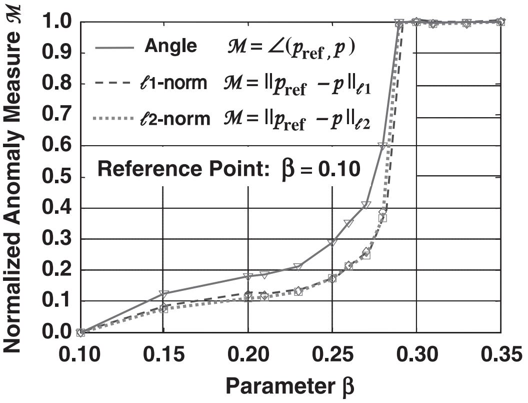


Fig. 5. Anomaly measure versus parameter β.

case, the reference point of nominal condition is represented by the parameter $\beta = 0.10$. The first plot, shown in solid line, shows $\mathscr{M}$ expressed as the angle (in radians) between the $\mathbf{p}$ vectors of the state machines under nominal and anomalous conditions, i.e., $\mathscr{M} = \angle(\mathbf{p}_{\mathrm{ref}},\ \mathbf{p}) \equiv \cos^{-1}(\frac{|\langle \mathbf{p}_{\mathrm{ref}}, \mathbf{p}\rangle|}{\|\mathbf{p}_{\mathrm{ref}}\|_{\ell 2}\|\mathbf{p}\|_{\ell 2}})$. The remaining two plots, one in dashed line and the other in dotted line, show the anomaly measure expressed as the ℓ_1-norm and ℓ_2-norm of the difference between the $\mathbf{p}$ vectors of the state machines under nominal and an anomalous conditions, i.e., $\mathscr{M} = \|\mathbf{p}_{\mathrm{ref}} - \mathbf{p}\|_{\ell 1}$ and $\mathscr{M} = \|\mathbf{p}_{\mathrm{ref}} - \mathbf{p}\|_{\ell 2}$, respectively. In each of the three plots, $\mathscr{M}$ is normalized to unity for better comparison. All three plots in Fig. 5 show gradual increase in the anomaly measure $\mathscr{M}$ for β in the approximate range of 0.10–0.25. At $\beta \approx 0.25$ and onwards, $\mathscr{M}$ starts increasing at a faster rate and finally saturates at $\beta \geqslant 0.29$. The large values of anomaly measure at $\beta = 0.29$ and beyond indicate the occurrence of period reduction as seen in Figs. 3 and 4. This abrupt disruption, preceded by gradual changes, is analogous to a phase transition in the thermodynamic sense [4], which can also be interpreted as a catastrophic disruption in a physical process. Hence, observation of modest changes in the anomaly measure may provide very early warnings for a forthcoming catastrophic failure as indicated by the gradual change in the $\beta - \mathscr{M}$ curve.

Following the steps (I1)–(I5) of the inverse problem in Section 5.1, the state probability vector $\mathbf{p}$ can be obtained for the stationary behavior under the known stimulus. The a priori information on the anomaly measure, generated in the step F6 of the forward problem in the Section 5.1, can then be used to determine the possible range in which β lies. Solutions of the forward problem can generate more information on different ranges of β under different input stimuli. Thus, the range of the unknown parameter β can be further narrowed down by repeating this step for other known stimuli as reported earlier [21]. This ensemble of information provides inputs for the inverse problem for detecting anomalies based on the sensor data collected in real time, during the operation of machineries.

7. Summary and conclusions

This paper presents a novel concept of anomaly detection in complex systems based on the tools of Symbolic Dynamics, Finite State Automata, and Pattern Recognition. It is assumed that dynamical systems under consideration exhibit nonlinear dynamical behavior on two time scales. Anomalies occur on a slow time scale that is (possibly) several orders of magnitude larger than the fast time scale of the system dynamics. It is also assumed that the unforced dynamical system (i.e., in the absence of external stimuli) is stationary at the fast time scale and that any non-stationary behavior is observable only on the slow time scale. This concept of small change detection in dynamical systems is elucidated on an active electronic circuit representing the forced Duffing equation with a slowly varying dissipation parameter. The time-series data of stationary phase trajectories are collected to create the respective symbolic dynamics (i.e., symbol sequences) using wavelet transform. The resulting state probability vector of the transition matrix is considered as the vector representation of a phase trajectory's stationary behavior. The distance between any two such vectors under the same stimulus is the measure of anomaly that the system has been subjected to. This vector representation of anomalies is more powerful than a scalar measure. The major conclusion of this research is that Symbolic Dynamics along with the stimulus-response methodology and having a vector representation of anomaly is effective for early detection of small anomalies.

The *D*-Markov machine, proposed for anomaly detection, is a suboptimal approximation of the ε-machine. It is important that this approximation is a sufficiently accurate representation of the nominal behavior. Research in this direction is in progress and the results will be presented in a forthcoming publication. Further theoretical research is recommended in the following areas:

- Separation of information-bearing part of the signal from noise.
- Identification of a relevant submanifold of the phase space and its partitioning to generate a symbol alphabet.
- Identification of appropriate wavelet basis functions for symbol generation and construction of a mapping from the wavelet space to the symbol space.
- Selection of the minimal *D* for the *D*-Markov machine and identification of the irreducible submatrix of the state transition matrix that contains relevant

information on anomalous behavior of the dynamical system.

Acknowledgements

The author wishes to thank Dr. Cosma Shalizi for making the CSSR code available and also for clarifying some of the underlying concepts of the ε-machine. The author is also thankful to Dr. Matthew Kennel for providing him with the software code on phase-space partitioning using symbolic false nearest neighbors. The author acknowledges technical contributions of Mr. Amit Surana and Mr. David Friedlander in developing the anomaly detection concept and of Mr. Venkatesh Rajagopalan for design and fabrication of the experimental apparatus on active electronic circuits.

Appendix A. Information theoretic quantities

This appendix introduces the concepts of standard information-theoretic quantities: *entropy rate*, *excess entropy* and *statistical complexity* [11], which are used to establish the anomaly measure in Section 5.

Entropy rate (h_μ): The entropy rate of a symbol string $\mathscr{S}$ is given by the Shannon entropy as follows:

$$h_\mu = \lim_{L\to\infty} \frac{H[L]}{L}, \tag{A.1}$$

where $H[L] \equiv -\sum_{s^L\in\mathscr{A}^L} P(s^L)\,\log_2(P(s^L))$ is the Shannon entropy of all L-blocks (i.e., symbol sequences of length L) in $\mathscr{S}$. The limit is guaranteed to exist for a stationary process [5]. The entropy rate quantifies the irreducible randomness in sequences produced by a source: the randomness that remains after the correlation and the structures in longer and longer sequence blocks are taken into account. For a symbol string $\mathscr{S}$ represented as an ε-machine, $h_\mu = H[\overrightarrow{S}^1|\mathscr{S}]$.

Excess entropy (**E**): The excess entropy of a symbol string $\mathscr{S}$ is defined as

$$\mathbf{E} = \sum_{L=1}^{\infty} [h_\mu(L) - h_\mu] \tag{A.2}$$

where $h_\mu(L) \equiv H[L]-H[L-1]$ is the estimate of how random the source appears if only L-blocks in $\mathscr{S}$ are considered. Excess entropy measures how much additional information must be gained about the sequence in order to reveal the actual per-symbol uncertainty h_μ, and thus measures difficulty in the prediction of the process. Excess entropy has alternate interpretations such as: it is the intrinsic redundancy in the process; geometrically it is a sub-extensive part of $H(L)$; and it represents how much historical information stored in the present is communicated to the future.

Statistical complexity (C_μ) [11]: The information of the probability distribution of causal states, as measured by Shannon entropy, yields the minimum average amount of memory needed to predict future configurations. This quantity is the *statistical complexity* of a symbol string $\mathscr{S}$, defined by Crutchfield and Young [7] as

$$C_\mu \equiv H(\mathscr{S}) = -\sum_{k=0}^{n-1} [Pr(S_k)\,\log_2 Pr(S_k)], \tag{A.4}$$

where n is the number of states of the finite-state machine constructed from the symbol string $\mathscr{S}$. As shown in [11], $\mathbf{E} \leqslant C_\mu$ in general, and $C_\mu = \mathbf{E} + Dh_\mu$.

Appendix B. Finite-type shift and sofic shift

This appendix very briefly introduces the concept of shift spaces with emphasis on finite shifts and sofic shifts that respectively characterize the D-Markov machine and the ε-machine described in the Section 4.4. The shift space formalism is a systematic way to study the properties of the underlying grammar, which represent the behavior of dynamical systems encoded through symbolic dynamics. The different shift spaces provide increasingly powerful classes of models that can be used to represent the patterns in the dynamical behavior.

Definition 2.1. Let $\mathscr{A}$ be a finite alphabet. The *full* $\mathscr{A}$*-shift* is the collection of all bi-infinite sequences of symbols from $\mathscr{A}$ and is denoted by:

$$\mathscr{A}^Z = \{x = (x_i)_{i\in Z}\colon x_i \in \mathscr{A}\ \forall i \in Z\}. \tag{A.5}$$

Definition 2.2. The *shift map* σ on the full shift $\mathscr{A}^Z$ maps a point x to a point $y=\sigma(x)$ whose ith coordinate is $y_i = x_{i+1}$.

A block is a finite sequence of symbols over $\mathscr{A}$. Let $x \in \mathscr{A}^Z$ and w be a block over $\mathscr{A}$. Then w occurs in x

if ∃ indices i and j such that $w = x_{[i,j]} = x_i x_{i+1} \cdots x_j$. Note that the empty block ε occurs in every x.

Let $\mathscr{F}$ be a collection of blocks, i.e., finite sequences of symbols over $\mathscr{A}$. Let $x \in \mathscr{A}^Z$ and w be a block over $\mathscr{A}$. Then w occurs in x if ∃ indices i and j such that $w = x_{[i,j]} = x_i x_{i+1} \cdots x_j$. For any such $\mathscr{F}$, let us define $X_{\mathscr{F}}$ to be the subset of sequences in $\mathscr{A}^Z$, which do not contain any block in $\mathscr{F}$.

Definition 2.3. A *shift space* is a subset X of a full shift $\mathscr{A}^Z$ such that $X = X_{\mathscr{F}}$ for some collection $\mathscr{F}$ of forbidden blocks over $\mathscr{A}$.

For a given shift space, the collection $\mathscr{F}$ is at most countable (i.e., finite or countably infinite) and is non-unique (i.e., there may be many such $\mathscr{F}$'s describing the shift space). As subshifts of full shifts, these spaces share a common feature called *shift invariance*. Since the constraints on points are given in terms of forbidden blocks alone and do not involve the coordinate at which a block might be forbidden, it follows that if $x \in X_{\mathscr{F}}$, then so are its shifts $\sigma(x)$ and $\sigma^{-1}(x)$. Therefore $\sigma(X_{\mathscr{F}}) = X_{\mathscr{F}}$, which is a necessary condition for a subset of $\mathscr{A}^Z$ to be a shift space. This property introduces the concept of shift dynamical systems.

Definition 2.4. Let X be a shift space and $\sigma_X : X \to X$ be the shift map. Then (X, σ_X) is known as a *shift dynamical system*.

The shift dynamical system mirrors the dynamics of the original dynamical system from which it is generated (by symbolic dynamics). Several examples of shift spaces are given in [15].

Rather than describing a shift space by specifying the forbidden blocks, it can also be specified by allowed blocks. This leads to the notion of a *language* of a shift.

Definition 2.5. Let X be a subset of a full shift, and let $\mathscr{B}_n(X)$ denote the set of all n-blocks (i.e., blocks of length n) that occur in X. The language of the shift space X is defined as:

$$\mathscr{B}(X) = \bigcup_{n=0}^{\infty} \mathscr{B}_n(X). \tag{A.6}$$

Sliding block codes: Let X be a shift space over $\mathscr{A}$, then $x \in X$ can be transformed into a new sequence $y = \cdots y_{-1} y_0 y_1 \cdots$ over another alphabet $\mathscr{U}$ as follows. Fix integers m and n such that $-m \leqslant n$. To compute y_i of the transformed sequence, we use a function Φ that depends on the "window" of coordinates of x from $i-m$ to $i+n$. Here $\Phi : \mathscr{B}_{m+n+1}(X) \to \mathscr{U}$ is a fixed *block map*, called a $(m+n+1)$-*block map* from the allowed $(m+n+1)$-blocks in X to symbols in $\mathscr{U}$. Therefore,

$$y_i = \Phi(x_{i-m} x_{i-m+1} \cdots x_{i+n}) = \Phi(x_{[i-m,i+n]}). \tag{A.7}$$

Definition 2.6. Let Φ be a block map as defined in Eq. (A.7). Then the map $\phi : X \to (\mathscr{U})^Z$ defined by $y = \phi(x)$ with y_i given by Eq. (A.7) is called the *sliding block code* with memory m and anticipation n induced by Φ.

Definition 2.7. Let X and Y be shift spaces, and $\phi : X \to Y$ be a sliding block code.

- If $\phi : X \to Y$ is onto, then ϕ is called a *factor code* from X onto Y.
- If $\phi : X \to Y$ is one-to-one, then ϕ is called an *embedding* of X into Y.
- If $\phi : X \to Y$ has a inverse (i.e., ∃ a sliding block code $\psi : Y \to X$ such that $\psi(\phi(x)) = x \ \forall x \in X$ and $\phi(\psi(y)) = y \ \forall \ y \in Y$), then ϕ is called a *conjugacy* from X to Y.

If ∃ a conjugacy from X to Y, then Y can be viewed as a copy of X, sharing all properties of X. Therefore, a conjugacy is often called topological conjugacy in literature.

Finite-type shifts: We now introduce the concept of finite-type shift that is the structure of the shift space in the D-Markov machine proposed in the Section 4.2.

Definition 2.8. A *finite-type shift* is a shift space that can be described by a finite collection of forbidden blocks (i.e., X having the form $X_{\mathscr{F}}$ for some finite set $\mathscr{F}$ of blocks).

An example of a finite shift is the *golden mean shift*, where the alphabet is $\Sigma = \{0, 1\}$ and the forbidden set $\mathscr{F} = \{11\}$. That is, $X = X_{\mathscr{F}}$ is the set of all binary sequences with no two consecutive 1's.

Definition 2.9. A finite-type shift is M-step or has memory M if it can be described by a collection of forbidden blocks all of which have length $M + 1$.

The properties of a finite-type shift are listed below:

- If X is a finite-type shift, then $\exists M \geqslant 0$ such that X is M-step.
- The language of the finite-type shift is characterized by the property that if two words overlap, then they can be glued together along their overlap to from another word in the language. Thus, a shift space X is an M-step finite-type shift iff whenever uv, $vw \in \mathscr{B}(X)$ and $|v| \geqslant M$, then $uvw \in \mathscr{B}(X)$.
- A shift space that is conjugate to a finite-type shift is itself a finite-type shift.
- A finite-type shift can be represented by a finite, directed graph and produces the collection of all bi-infinite walks (i.e. sequence of edges) on the graph.

Sofic shifts: The sofic shift is the structure of the shift space in the ε-machines [7,20] in Section 4.1. Let us label the edges of a graph with symbols from an alphabet $\mathscr{A}$, where two or more edges are allowed to have the same label. Every bi-infinite walk on the graph yields a point in $\mathscr{A}^Z$ by reading the labels of its edges, and the set of all such points is called a *sofic shift*.

Definition 2.10. A graph G consists of a finite set $\mathscr{V} = \mathscr{V}(G)$ of vertices together with a finite set $\mathscr{E} = \mathscr{E}(G)$ of edges. Each edge $e \in \mathscr{E}(G)$ starts at a vertex denoted by $i(e) \in \mathscr{V}(G)$ and terminates at a vertex $t(e) \in \mathscr{V}(G)$ (which can be the same as $i(e)$). There may be more that one edge between a given initial state and terminal state; a set of such edges is called a set of multiple edges. An edge e with $i(e) = t(e)$ is called a self-loop.

Definition 2.11. A *labelled graph* $\mathscr{G}$ is a pair $(G, \mathscr{L})$, where G is a graph with edge set $\mathscr{E}$, and $\mathscr{L} : \mathscr{E} \to \mathscr{A}$ assigns a label $\mathscr{L}(e)$ to each edge e of G from the finite alphabet $\mathscr{A}$. The *underlying graph* of $\mathscr{G}$ is G.

Definition 2.12. A subset X of a full shift is a *sofic shift* if $X = X_{\mathscr{G}}$ for some labelled graph $\mathscr{G}$. A *presentation* of a sofic shift X is a labelled graph $\mathscr{G}$ for which $X_{\mathscr{G}} = X$.

An example of a sofic shift is the *even shift*, which is the set of all binary sequences with only even number of 0’s between any two 1’s. That is, the forbidden set $\mathscr{F}$ is the collection $\{10^{2n+1}\colon n \geqslant 0\}$.

Some of the salient characterization of sofic shifts are presented below [15]:

- Every finite-type shift qualifies as a sofic shift.
- A shift space is sofic iff it is a factor of a finite-type shift.
- The class of sofic shifts is the smallest collection of shifts spaces that contains all finite-type shifts and also contains all factors of each space in the collection.
- A sofic shift that does not have finite-type subshifts is called a *strictly sofic*. For example, the *even shift* is strictly sofic [15].
- A factor of a sofic shift is a sofic shift.
- A shift space conjugate to a sofic shift is itself sofic.
- A distinction between finite-type shifts and sofic shifts can be characterized in terms of the *memory*. While finite-type shifts use finite-*length* memory, sofic shifts require finite *amount* of memory. In contrast, context-free shifts require infinite amount of memory [12].

References

[1] H.D.I. Abarbanel, The Analysis of Observed Chaotic Data, Springer, New York, 1996.

[2] R. Badii, A. Politi, Complexity Hierarchical Structures and Scaling in Physics, Cambridge University Press, UK, 1997.

[3] R.B. Bapat, T.E.S. Raghavan, Nonnegative Matrices and Applications, Cambridge University Press, Cambridge, 1997.

[4] C. Beck, F. Schlogl, Thermodynamics of Chaotic Systems: an Introduction, Cambridge University Press, UK, 1993.

[5] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley, New York, 1991.

[6] J.P. Crutchfield, The calculi of emergence: Computation, dynamics and induction, Physica D 75 (1994) 11–54.

[7] J.P. Crutchfield, K. Young, Inferring statistical complexity, Phys. Rev. Lett. 63 (1989) 105–108.

[8] R.L. Davidchack, Y.C. Lai, E.M. Bolt, H. Dhamala, Estimating generating partitions of chaotic systems by unstable periodic orbits, Phys. Rev. E 61 (2000) 1353–1356.

[9] R. Duda, P. Hart, D. Stork, Pattern Classification, Wiley, New York, 2001.

[10] D.P. Feldman, Computational mechanics of classical spin systems, Ph.D. Dissertation, University of California, Davis, 1998.

[11] D.P. Feldman, J.P. Crutchfield, Discovering non-critical organization: statistical mechanical, information theoretic, and computational views of patterns in one-dimensional spin systems, Santa Fe Institute Working Paper 98-04-026, 1998.

[12] H.E. Hopcroft, R. Motwani, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, 2nd Edition, Addison-Wesley, Boston, 2001.

[13] M.B. Kennel, M. Buhl, Estimating good discrete partitions form observed data: symbolic false nearest neighbors, http://arxiv.org/PS_cache/nlin/pdf/0304/0304054.pdf, 2003.

[14] B.P. Kitchens, Symbolic Dynamics: One Sided, Two sided and Countable State Markov Shifts, Springer, New York, 1998.

[15] D. Lind, M. Marcus, An Introduction to Symbolic Dynamics and Coding, Cambridge University Press, UK, 1995.

[16] M. Markou, S. Singh, Novelty detection: a review—parts 1 and 2, Signal Processing 83 (2003) 2481–2521.

[17] A.W. Naylor, G.R. Sell, Linear Operator Theory in Engineering and Science, Springer, New York, 1982.

[18] E. Ott, Chaos in Dynamical Systems, Cambridge University Press, UK, 1993.

[19] R.K. Pathria, Statistical Mechanics, 2nd Edition, ButterworthHeinemann, Oxford, UK, 1998.

[20] C.R. Shalizi, K.L. Shalizi, J.P. Crutchfield, An algorithm for pattern discovery in time series, SFI Working Paper 02-10-060, 2002.

[21] A. Surana, A. Ray, S.C. Chin, Anomaly detection in complex systems, in: Fifth IFAC Symposium on Fault Detection, Supervision and Safety of Technical Process, Washington, DC, 2003.

[22] F. Takens, Detecting strange attractors in turbulence, in: D. Rand, L.S. Young (Eds.), Proceedings of the Symposion Dynamical Systems and Turbulence, Warwick, 1980, Lecture Notes in Mathematical, Vol. no 898, Springer, Berlin, 1981, p. 366.

[23] J.M.T. Thompson, H.B. Stewart, Nonlinear Dynamics and Chaos, Wiley, Chichester, UK, 1986.

[24] D.R. Upper, Theory and algorithms for hidden Markov models and generalized hidden Markov models, Ph.D. Dissertation in Mathematics, University of California, Berkeley, 1997.

[25] H. Zang, A. Ray, S. Phoha, Hybrid life extending control of mechanical systems: experimental validation of the concept, Automatica 36 (2000) 23–36.

Available online at www.sciencedirect.com

ScienceDirect

Signal Processing 86 (2006) 3309–3320

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

Symbolic time series analysis via wavelet-based partitioning☆

Venkatesh Rajagopalan, Asok Ray*

The Pennsylvania State University, University Park, PA 16802, USA

Received 24 March 2005; accepted 30 January 2006
Available online 28 February 2006

Abstract

Symbolic time series analysis (*STSA*) of complex systems for anomaly detection has been recently introduced in literature. An important feature of the *STSA* method is extraction of relevant information, imbedded in the measured time series data, to generate symbol sequences. This paper presents a wavelet-based partitioning approach for symbol generation, instead of the currently practiced method of phase-space partitioning. Various aspects of the proposed technique, such as wavelet selection, noise mitigation, and robustness to spurious disturbances, are discussed. The wavelet-based partitioning in *STSA* is experimentally validated on laboratory apparatuses for anomaly/damage detection. Its efficacy is investigated by comparison with phase-space partitioning.

Keywords: Symbolic time series analysis; Wavelets; Fault detection

1. Introduction

The concept of symbolic time series analysis (*STSA*) has been recently proposed for anomaly detection in complex systems [1]. Several case studies [2–5] in anomaly detection show that *STSA* can be more effective than existing pattern recognition techniques (e.g., principal component analysis and neural networks). The *STSA* method has also been demonstrated for fault detection in electromechanical systems, such as three-phase induction motors [6] and helical gearbox in rotorcraft [7].

A crucial step in *STSA* is extraction of relevant information, imbedded in the measured time series data, to generate symbol sequences. Symbol generation requires partitioning of the data space to obtain the symbol sequences [8,9]. Various partitioning techniques have been suggested in literature for symbol generation, which include variance-based [10], entropy-based [11], and hierarchical clustering [12] methods. A survey of clustering techniques is provided in [13]. In addition to these methods, another scheme of partitioning, based on *symbolic false nearest neighbors* (*SFNN*), was reported by Kennel and Buhl [14]. The objective of *SFNN* partitioning is to ensure that points that are close to each other in the symbol space are also close to each other in the phase space. Partitions that yield a smaller proportion of *SFNN* are considered optimal. However, this partitioning method may become computationally

☆This work has been supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under Grant No. DAAD19-01-1-0646.

*Corresponding author. Tel.: +1 814 865 63 77; fax: +1 814 863 48 48.

E-mail addresses: vxr139@psu.edu (V. Rajagopalan), axr2@psu.edu (A. Ray).

0165-1684/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2006.01.014

very expensive if the dimension of the phase space is large or if the data set is contaminated by noise, since noise induces false symbols.

Partitioning of wavelet-transformed data potentially alleviates the above-mentioned shortcomings and is particularly effective with noisy data from high-dimensional dynamical systems. Usage of wavelet transform for symbolization has been recently introduced by the authors [1,15]. This paper elaborates the concept of wavelet-based partitioning for *STSA* and its major features are delineated below.

- Selection of the wavelet basis and scale range.
- Noise mitigation in the measured time series data prior to symbol generation.
- Robustness of extracted information from symbol sequences.
- Enhancement of computational efficiency for anomaly detection.
- Validation of the *STSA* method on laboratory apparatuses for anomaly/damage detection.

The paper is organized into six sections including the present section. Section 2 focuses on key technical aspects of wavelet analysis such as selection of wavelet basis and scales. Section 3 presents the maximum entropy method of partitioning the space of wavelet coefficients. Section 4 elucidates the underlying principles of wavelet-based partitioning via simulation examples. Section 5 presents experimental results on laboratory apparatuses to demonstrate efficacy of wavelet-based *STSA* for anomaly detection. Section 6 summarizes the paper and makes major conclusions along with recommendations for future research.

2. Wavelet analysis of time series data

This section presents generation of wavelet coefficients from measured time series data, and their arrangement for symbol generation. Specifically, issues of wavelet basis and scale range selection are addressed.

Preprocessing of time series data is often necessary for extraction of pertinent information. Fourier analysis is sufficient if the signal to be analyzed is stationary and if the time period is accurately known. However, Fourier analysis may not be appropriate if the signal has non-stationary characteristics such as drifts, abrupt asynchronous changes, and frequency trends. Wavelet analysis alleviates these difficulties via adaptive usage of long windows for retrieving low-frequency information and short windows for high-frequency information [16,17]. The ability to perform flexible localized analysis is one of the striking features of wavelet transform.

In multi-resolution analysis (*MRA*) of wavelet transform, a continuous signal $f \in \mathbb{H}$, where $\mathbb{H}$ is a Hilbert space, is decomposed as a linear combination of time translations of scaled versions of a suitably chosen scaling function $\phi(t)$ and the derived wavelet function $\psi(t)$. Let the sequence $\{\phi_{j,k}\}$ belong to another Hilbert space $\mathbb{M}$ with a countable measure, where the scale $s = 2^j$ and time translation $\tau = 2^{-j}k$. If the sequence $\{\phi_{j,k}\}$ is a frame for the Hilbert space $\mathbb{H}$ with a frame representation operator $\mathbb{L}$, then there are positive real scalars A and B such that

$$A\|f\|^2_{\mathbb{H}} \leqslant \|\mathbb{L}f\|^2_{\mathbb{M}} \leqslant B\|f\|^2_{\mathbb{H}} \quad \forall f \in \mathbb{H}, \tag{1}$$

where $\mathbb{L}f = \{\langle f, \phi_{j,k}\rangle\}$ and $\|\mathbb{L}f\|_{\mathbb{M}}$ is an appropriate norm, e.g., $\|\mathbb{L}f\|_{\mathbb{M}} = \sqrt{\sum_j \sum_k |\langle f, \phi_{j,k}\rangle|^2}$ is a candidate norm; and $\langle x, y\rangle$ is the inner product of x and y, both belonging to $\mathbb{H}$.

The above relationship is a norm equivalence and represents the degree of coherence of the signal f with respect to the frame set of scaling functions; it may be interpreted as enforcing an approximate energy transfer between the domains $\mathbb{H}$ and $\mathbb{L}(\mathbb{H})$. In other words, for all signals $f \in \mathbb{H}$, a scaled amount of energy is distributed in the coefficient domain where the scale factor lies between A and B [16]. However, the energy distribution is dependent on the signal's degree of coherence with the underlying frame $\{\phi_{j,k}\}$. For a signal f, which is coherent with respect to the frame $\{\phi_{j,k}\}$, norm equivalence in the frame representation necessarily implies that a few coefficients contain most of the signal energy and hence have relatively large magnitudes. Similarly, pure noise signal w being incoherent with respect to the set $\{\phi_{j,k}\}$, must have a frame representation in which the noise energy is spread out over a large number of coefficients. Consequently, these coefficients have a relatively small magnitude [17].

Let $\tilde{f}$ be a noise corrupted version of the original signal f expressed as

$$\tilde{f} = f + \sigma w, \tag{2}$$

where w is additive white gaussian noise with zero mean and unit variance and σ is the noise level.

Then, the inner product of $\tilde{f}$ and $\phi_{j,k}$ is obtained as

$$\langle \tilde{f}, \phi_{j,k} \rangle = \underbrace{\langle f, \phi_{j,k} \rangle}_{\text{signal part}} + \sigma \underbrace{\langle w, \phi_{j,k} \rangle}_{\text{noise part}}. \quad (3)$$

The noise part in Eq. (3) may further be reduced if the scales over which coefficients are obtained are properly chosen.

For every wavelet, there exists a certain frequency called the center frequency F_c that has the maximum modulus in the Fourier transform of the wavelet [18]. The pseudo-frequency f_p of the wavelet at a particular scale α is given by the following formula [18,19]:

$$f_p = \frac{F_c}{\alpha \Delta t}, \quad (4)$$

where Δt is the sampling interval. Fig. 1 depicts the center frequency associated with the Daubechies 4 wavelet 'db4' [16,20].

The power spectral density (*PSD*) of the signal provides the information about the frequency content of the signal. This information along with Eq. (4) can be used for scale selection. The procedure of selecting the scales is summarized below:

- Identification of the frequencies of interest through *PSD* analysis of time series data.
- Substitution of the above frequencies in place of f_p in Eq. (4) to obtain the respective scales in terms of the known parameters F_c and Δt.

The wavelet coefficients of the signal are significantly large when the pseudo-frequency f_p of the wavelet corresponds to the locally dominant frequencies in the underlying signal. Example 1 in Section 4 illustrates how the choice of wavelet and scale affect the coefficients. Examples 2 and 3 illustrate noise suppression and robustness. Three experimental studies in Section 5 illustrate enhancement of anomaly/damage detection using *STSA* and make comparisons of wavelet-based partitioning and *SFNN* phase-space partitioning in this regard.

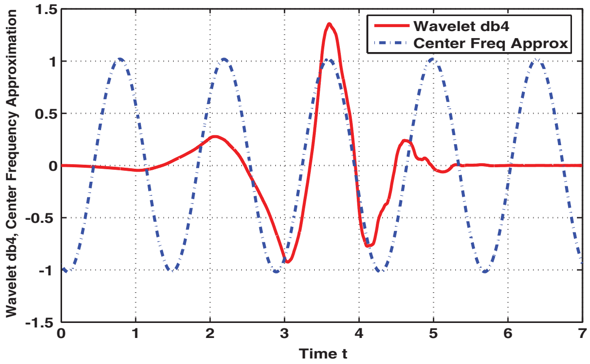


Fig. 1. Center frequency approximation for wavelet db4.

Upon selection of the wavelet basis and scale range, the wavelet coefficients are obtained. These coefficients are stacked at selected time-shift positions, starting with the smallest value of scale and ending with its largest value and then back from the largest value to the smallest value of the scale at the next instant of time shift. In the sequel, this one-dimensional array of arranged wavelet coefficients is called the *scale series* data, which is structurally similar to time series data in the phase space. For symbol generation, the scale series data can be handled in a similar way as time series data.

3. Symbolization of scale series data

This section presents symbolization of wavelet coefficients by maximum entropy partitioning. The scale series data are partitioned to construct the symbol alphabet and to generate symbol sequences.

In the wavelet-based partitioning scheme, as introduced in an earlier publication [1], the maximum and minimum of the scale series are evaluated and the ordinates between the maximum and minimum are divided into equal-sized regions. These regions are mutually disjoint and thus form a partition. Each region is then labelled with one symbol from the alphabet. If the data point lies in a particular region, it is coded with the symbol associated with that region. Thus, a sequence of symbols is created from a given sequence of scale series data. This type of partitioning is called uniform partitioning in the sequel. The left-hand plot in Fig. 2 depicts uniform partitioning of noise contaminated signal $\sin(2\pi t)$ with alphabet size $|\Sigma| = 4$. Note that the partition segments are of equal size.

Intuitively, it is more reasonable if the information-rich regions of the data set are partitioned finer and those with sparse information are partitioned coarser. To achieve this objective, a partitioning method is adopted such that the entropy of the generated symbol sequence is maximized [15]. Maximum entropy partitioning is abbreviated as *ME* partitioning in the sequel. The procedure for obtaining an *ME* partition is described below.

Let N be the length of the data set and $|\Sigma|$ be the size of the symbol alphabet (i.e., the number of

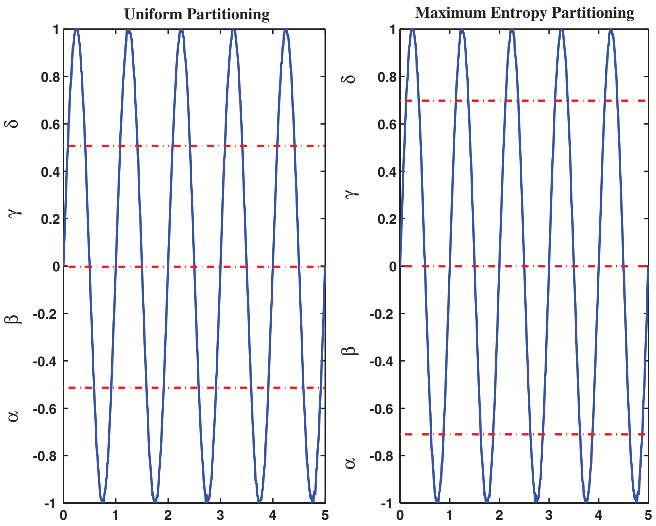


Fig. 2. Examples of uniform and ME partitioning with $|\Sigma| = 4$.

disjoint elements in the partition). The data set is sorted in ascending order. Starting from the first point in the sorted data, every consecutive data segment of length $\lfloor N/|\Sigma| \rfloor$ forms a distinct element of the partition, where $\lfloor x \rfloor$ represents the greatest integer less than or equal to x.

The right-hand plot in Fig. 2 shows *ME* partitioning for the noise-contaminated signal $\sin(2\pi t)$ with $|\Sigma| = 4$. As expected, the size of the partitions are not equal, but the probabilities of the symbols are equal. Variations in data patterns are more likely to be reflected in the symbol sequence obtained under *ME* partitioning than other partitioning.

The choice of the alphabet size $|\Sigma|$ plays a crucial role in STSA. For example, a small value of $|\Sigma|$ may prove inadequate for capturing the characteristics of the scale series data. On the other hand, a large value of $|\Sigma|$ may lead to redundancy and waste of computational resources.

Selection of the alphabet size $|\Sigma|$ is an area of active research; an entropy-based approach has been adopted for selecting $|\Sigma|$ in this paper. Let $H(k)$ denote the Shannon entropy of the symbol sequence obtained by partitioning the data set with k symbols:

$$H(k) = -\sum_{i=1}^{i=k} p_i \log_2 p_i, \tag{5}$$

where p_i represents the probability of occurrence of the symbol σ_i. Note that $H(1) = 0$ because $p_i = 0$ or 1 with $i = 1$. If the underlying data set has sufficient information content, then the entropy achieved under *ME* partitioning would be $\log_2(k)$, which corresponds to the uniform distribution. We define a quantity $h(\cdot)$ to represent the change in entropy with respect to the number $|\Sigma|$ of symbols as

$$h(k) \triangleq H(k) - H(k-1) \quad \forall k \geqslant 2. \tag{6}$$

The algorithm for alphabet size selection is given below.

Step 1: Set $k = 2$. Choose a threshold ε_h, where $0 < \varepsilon_h \ll 1$.

Step 2: Sort the data set (of length N) in the ascending order.

Step 3: Every consecutive segment of length $\lfloor N/k \rfloor$ in the sorted data set (of length N) forms a distinct element of the partition.

Step 4: Convert the raw data into a symbol sequence with the partitions obtained in Step 3. If the data point lies within or on the lower bound of a partition, it is coded with the symbol associated with that partition.

Step 5: Compute the symbol probabilities p_i, i = 1,2,…, k.

Step 6: Compute $H(k) = -\sum_{i=1}^{i=k} p_i \log_2 p_i$ and $h(k) = H(k) - H(k-1)$.

Step 7: If $h(k) < \varepsilon_h$, then exit; else increment k by 1 and go to Step 3.

In general, a small ε_h leads to a large size of the symbol alphabet, resulting in increased computation. Also, a larger alphabet makes the partitioning finer. This might increase the probability of false symbols being induced by noise. On the other hand, a large ε_h leads to a small alphabet size that may prove inadequate for capturing the pertinent information. Hence, there is a trade-off between accuracy and computational speed when ε_h is chosen. The variance of the noise process associated with the signal may serve as a guideline for selection of ε_h.

4. Validation of wavelet-based partitioning

This section presents simulation cases to validate symbolization of measured time series data via partitioning of the wavelet coefficients. The underlying concepts are illustrated by three examples.

4.1. Example 1: choice of wavelet parameters

This example illustrates how the choice of wavelet basis and scale range affects the coefficients that, in turn, determine symbol generation for anomaly detection [1]. Let us consider the following sinusoidal signal:

$$y(t) = \cos(2\pi t) \quad \forall t \in [-5, +5]. \tag{7}$$

The frequency of $y(t)$ in Eq. (7) is 1.00 Hz. The Gaussian wavelet 9 (‘gaus9’) (see [20, p. 74]) matches the shape of $y(t)$ well, as seen in Fig. 3 that compares an appropriately scaled and translated version of the ‘gaus9’ wavelet with the signal $y(t)$.

To demonstrate the impact of the chosen wavelet parameters on the coefficients, the wavelet basis ‘db1’ is also considered for comparison purposes. The signal $y(t)$ is sampled at 100 Hz (i.e., the sampling interval $\Delta t = 0.01$ s). The wavelet coefficients of the signal $y(t)$ are obtained for various scales with both wavelets, ‘gaus9’ and ‘db1’. The norm of the coefficients corresponding to each scale and the pseudo-frequencies of the wavelet corresponding to the chosen scales are calculated. Fig. 4 shows the plot of the norm of coefficients and the pseudo-frequencies of the wavelet.

It is observed in Fig. 4 that, for both wavelets ‘gaus9’ and ‘db1’, the maximum of the norm is obtained at $f_{\rm p} \approx 1.00$ Hz. In fact, it is exactly at 1.00 Hz for ‘gaus9’. Furthermore, the value of the peak norm achieved with ‘gaus9’ is appreciably greater than that with wavelet ‘db1’. In other words, the coefficients obtained with ‘gaus9’ are more significant than those obtained with ‘db1’. Another observation is that the norm curve for ‘gaus9’ shows a greater rate of decay across pseudo-frequencies than that of ‘db1’. More energy is concentrated in a narrow band frequencies around 1.00 Hz in the case of ‘gaus9’. These observations imply that high energy compaction can be achieved with fewer coefficients if the wavelet and the scales are chosen as stated in Section 2. A favorable implication of fewer coefficients is fewer number of symbols for analysis and hence an improvement in computational efficiency.

4.2. Example 2: noise suppression

This example demonstrates how noise suppression is achieved with wavelets. Let the signal $y(t)$ in Eq. (7) be corrupted with additive zero-mean white Gaussian noise $w(t)$,

$$\tilde{y}(t) = y(t) + \sigma w(t). \tag{8}$$

A common measure of noise in a noise-corrupted signal is the signal-to-noise ratio (SNR) that is

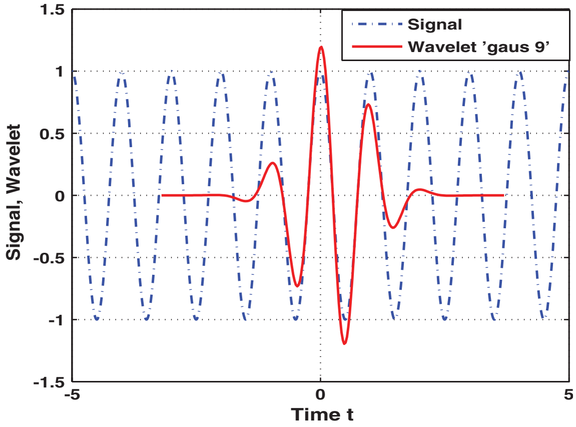


Fig. 3. Comparison of the wavelet basis ‘gaus9’ and the signal.

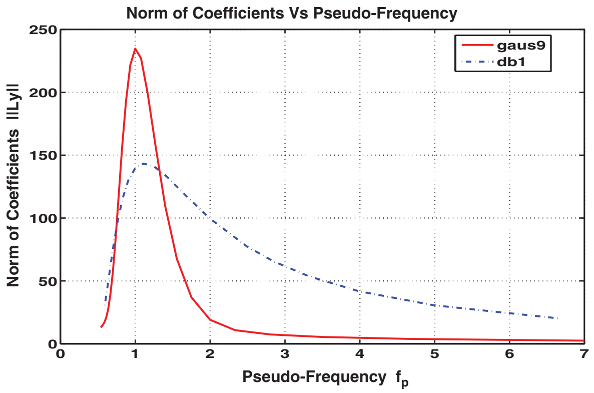


Fig. 4. Coefficient norm and pseudo-frequency for different wavelets.

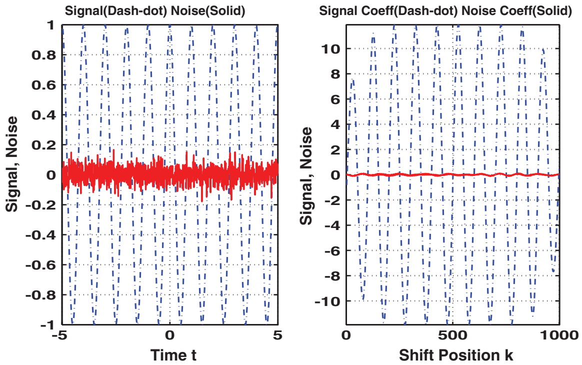


Fig. 5. Signal and noise profiles at $\sigma = 0.05$.

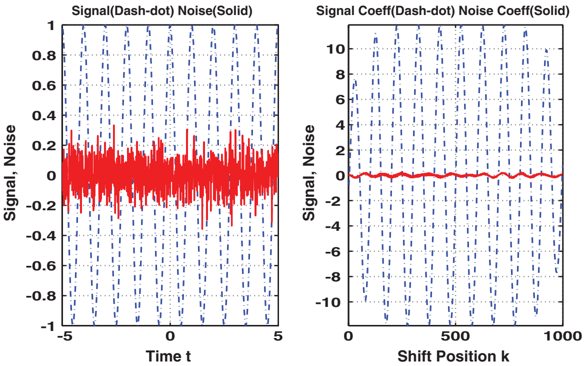


Fig. 6. Signal and noise profiles at $\sigma = 0.10$.

defined as:

$$\text{SNR} \triangleq \frac{\|y\|_{\mathbb{H}}^2}{\|\sigma\, w\|_{\mathbb{H}}^2}, \tag{9}$$

where y and w are functions of time. Similar to Eq. (9), the SNR in the wavelet domain is defined as

$$\widetilde{\text{SNR}} \triangleq \frac{\|\mathbb{L}y\|_{\mathbb{M}}^2}{\|\sigma\, \mathbb{L}w\|_{\mathbb{M}}^2}, \tag{10}$$

where $\mathbb{L}y$ and $\mathbb{L}w$, which belong to the Hilbert space $\mathbb{M}$ (see Section 2), represent the wavelet coefficients of the signal y and the noise w.

Numerical experiments have been performed with $\sigma \in \{0.05, 0.1\}$. The signal is sampled at 100 Hz (i.e., $\Delta t = 0.01$ s). The scales are determined following Eq. (4), such that the pseudo-frequency of the wavelet matches the frequency of the signal. Fig. 5 depicts the time domain plot (left plate) and coefficient plot (right plate) of the signal y and white Gaussian noise having standard deviation $\sigma = 0.05$. Similarly, Fig. 6 depicts the time domain plot (left plate) and coefficient plot (right plate) of the signal y and white Gaussian noise having standard deviation $\sigma = 0.10$. Table 1 lists the values of SNR and $\widetilde{\text{SNR}}$, averaged over 20 simulation runs.

Both Figs. 5 and 6 show that $\widetilde{\text{SNR}}$ is significantly larger than SNR. That is, the wavelet-transformed signal is significantly de-noised relative to the time domain signal. This is expected because the noise is incoherent with the wavelet while the signal enjoys a great degree of coherence with the wavelet. Thus, symbols generated from wavelet coefficients would reflect the characteristics of the signal with more fidelity than those obtained with time domain signals.

Table 1
SNR values

	$\sigma = 0.05$	$\sigma = 0.1$
SNR	191.55	50.89
$\widetilde{\text{SNR}}$	25195	4281.5

4.3. *Example 3: robustness of symbol probability vector*

The symbolization scheme is developed to enhance real-time detection of slowly varying anomalies in dynamical systems [1]. Of critical importance is the symbol probability vector $\mathbf{p}$ whose elements denote the probability of occurrence of individual symbols in the symbol sequence. The vector $\mathbf{p}$ must be robust relative to measurement noise and spurious disturbances and, at the same time, be sensitive enough to detect small slowly varying anomalies from the observed data set. A distortion measure for the symbol probability vector is introduced below

$$\delta_t \triangleq \|p_t - \tilde{p}_t\|_1, \tag{11}$$

where the subscript t denotes that the probability vectors correspond to symbols generated from time domain signals; and $\| \bullet \|_1$ is the sum of the absolute values of the elements of the vector $\bullet$. The vector p_t, with $\|p_t\|_1 = 1$, corresponds to the uncorrupted signal and $\tilde{p}_t$ corresponds to the corrupted signal. Similar to Eq. (11), distortion measure in the wavelet scale domain is defined as

$$\delta_{\text{s}} \triangleq \|p_{\text{s}} - \tilde{p}_{\text{s}}\|_1, \tag{12}$$

where the subscript s denotes that the probability vectors correspond to symbols generated from wavelet scale domain signals (i.e., scale series data). Therefore, lower is the distortion ratio, closer is the probabilistic representation of the corrupted signal to that of the uncorrupted signal, i.e., greater is the robustness to noise and spurious disturbances.

The partitions are obtained, in case of time domain, by employing the maximum entropy criterion on the time series data of the signal. In the wavelet domain, the partitions are obtained with the corresponding scale series data, as defined in Section 2. In both time domain and wavelet domain, the probability vectors p and $\tilde{p}$ are computed with the same partitions for the uncorrupted and corrupted signals.

The symbol alphabet size $|\Sigma|$ is chosen to be 4 in this example. The partitions are obtained as mentioned before for the signal y and its transform, i.e., the coefficient vector $\mathbb{L}y$. Table 2 lists the values of distortion ratios δ_t and δ_s, averaged over 20 simulation runs.

It is seen that distortion measures are far smaller in the wavelet scale domain than those in the time domain. This observation implies that the symbol probabilities are significantly more robust to measurement noise and spurious disturbances in the wavelet domain than in the time domain. Hence, it may be inferred that symbols generated from the wavelet coefficients would be better for anomaly detection as the effects of noise to induce errors in the symbol probabilities are significantly mitigated.

5. Experimental results on anomaly detection

This section evaluates the performance of wavelet-based partitioning in *STSA* for anomaly/damage detection. This is demonstrated via experimentation on the following laboratory apparatuses:

- Anomaly detection on a nonlinear electronic system apparatus [1].
- Structural damage detection on a mechanical vibration system apparatus [3].
- Damage detection in polycrystalline alloys on a fatigue testing apparatus [5].

Table 2
Distortion ratios

	$\sigma = 0.05$	$\sigma = 0.1$
δ_t	0.040	0.054
δ_s	0.006	0.010

5.1. Anomaly detection in nonlinear systems

This example demonstrates efficacy of the *STSA* method for anomaly detection in nonlinear systems. Experiments have been conducted on a laboratory apparatus [1] that emulates the forced Duffing equation [21], modelled as

$$\frac{d^2y}{dt^2} + \beta\frac{dy}{dt} + y(t) + y^3(t) = A\cos(\Omega t), \tag{13}$$

where the dissipation parameter β varies slowly with respect to the response of the dynamical system; $\beta = 0.1$ represents the nominal condition; and a change in the value of β is considered as an anomaly. With amplitude $A = 22.0$ and $\Omega = 5.0$, a sharp change in the behavior is noticed around $\beta = 0.29$, possibly due to bifurcation. The phase plots and time-response plots, depicting this drastic change behavior, are not presented here as they are provided in an earlier publication [1].

The objective of anomaly detection is to identify small changes in the parameter β as early as possible and well before it manifests a drastic change in the system dynamics. The details of the experimental apparatus are provided in [22]. Time series data of the signal $y(t)$ from the experimental apparatus is used for symbolic analysis.

The first step in the analysis is selection of the wavelet basis. The time series data of the signal and a scaled and translated version of the wavelet ‘gaus1’ are shown in the left-hand plate of Fig. 7. For the purpose of comparison, the right-hand plate of Fig. 7 shows the same time series data of the signal and a scaled and translated version of the wavelet ‘db1’ that was used in [1] for wavelet analysis. Since ‘gaus1’ matches the shape of the signal more closely than ‘db1’, the wavelet ‘gaus1’ is

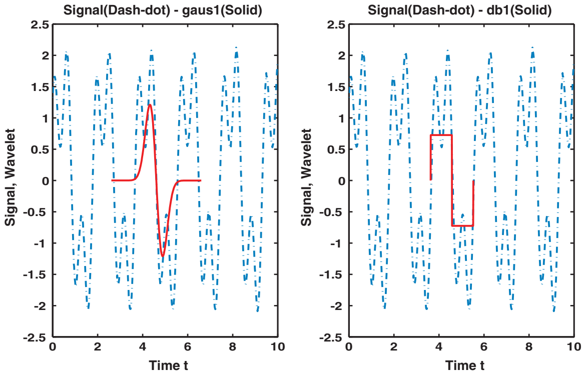


Fig. 7. Profiles of signal and wavelet.

better suited for STSA of this data sequence than 'db1'. Once the wavelet is chosen, the next step is identification of the frequencies of interest. The top and bottom plates of Fig. 8 depict the power spectra of the system for $\beta = 0.1$ and 0.29, respectively.

Fig. 8 shows that frequencies in the neighborhood of 0.54 Hz, though present at the nominal condition $\beta = 0.10$, are absent at $\beta = 0.29$. The wavelet coefficients at scales, corresponding to the pseudo-frequency of 0.54 Hz, would be smaller in magnitude in the anomalous condition(s) when compared with those in the nominal condition. Hence, by choosing scales that correspond to pseudo-frequencies around 0.54 Hz, early detection can be achieved more effectively.

The next aspect of anomaly detection via STSA is the choice of number of symbols, i.e., cardinality $|\Sigma|$ of the symbol alphabet Σ. The scale series data, at the nominal condition, is partitioned into a symbol sequence starting with $|\Sigma| = 2$ and the threshold parameter ε_h is chosen to be 0.2. Fig. 9 depicts the change in entropy h versus the number of symbols $|\Sigma|$. It is seen that h monotonically decreases with $|\Sigma|$ and becomes less than ε_h when $|\Sigma| = 8$. Accordingly, the number of symbols $|\Sigma|$ is chosen to be eight. A smaller value of ε_h results in increased number of symbols, which would increase computation with (possibly) no significant gain in accuracy of anomaly detection.

The partition is obtained using data obtained under the nominal ($\beta = 0.1$) condition. Once the partition is generated, it remains invariant. As the dynamical behavior of the system changes due to variations in β, the statistical characteristics of the symbol sequences are also altered and so do the symbol probabilities. A measure could be induced on the symbol probability vectors obtained under different anomalous conditions, to quantify these changes. Such a measure is called the anomaly measure $\mathbf{M}$. The metric $\mathbf{M_k} = d(\mathbf{p_0}, \mathbf{p_k})$ is an anomaly measure, where $\mathbf{p_0}$ and $\mathbf{p_k}$ represent the symbol probability vectors under nominal and anomalous conditions, respectively. A candidate anomaly measure is the angle between the symbol probability vectors under nominal and anomalous conditions. This measure is defined as

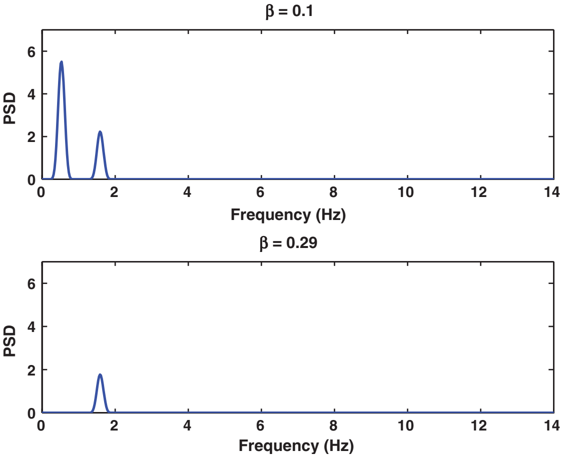


Fig. 8. Power spectrum plots at nominal and anomalous conditions.

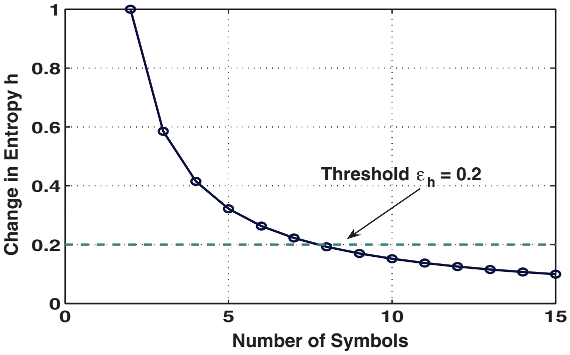


Fig. 9. Selection of number of symbols from change in entropy.

$$\mathbf{M_k} = \arccos\left(\frac{\langle \mathbf{p_0}, \mathbf{p_k} \rangle}{\|\mathbf{p_0}\|_2 \|\mathbf{p_k}\|_2}\right), \tag{14}$$

where $\langle x, y \rangle$ is the inner product of the vectors x and y; and $\|x\|_2$ is the Euclidean norm of x.

Fig. 10 compares the profiles of the anomaly measures $\mathbf{M_k}$ in Eq. (14) obtained with wavelet 'gaus1' under ME partitioning and uniform partitioning. With β increasing from 0.1, there is a gradual increase in the anomaly measure much before the abrupt change in the vicinity of $\beta = 0.29$ takes place. This indicates growth of the anomaly even before any notable change in the dynamical behavior takes place. At this point, the anomaly measure starts increasing relatively more rapidly suggesting the onset of a forthcoming catastrophic failure. Under ME partitioning, the larger values of the anomaly measure at smaller values of β and gradual increase in both slope and curvature of the anomaly measure curve would facilitate anomaly detection significantly before it is possible to do so under uniform partitioning. Therefore, with regard to early detection of anomalies, ME partitioning appears to be more effective than uniform partitioning.

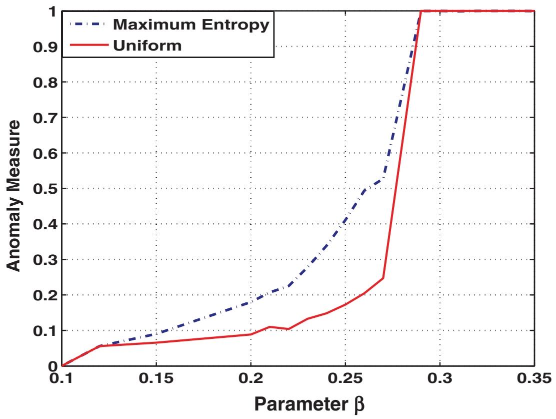


Fig. 10. Maximum entropy partitioning and uniform partitioning.

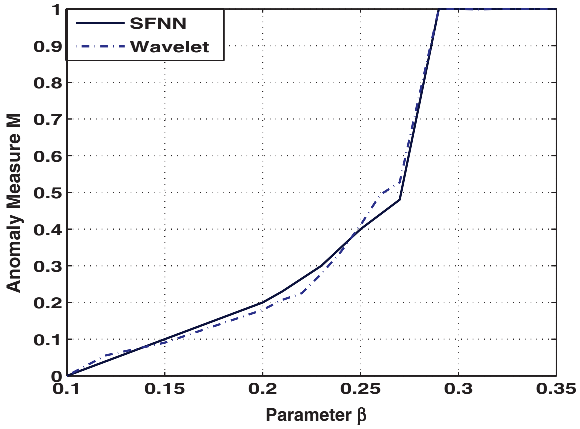


Fig. 11. Anomaly detection on the electronic system apparatus.

Fig. 11 depicts the anomaly measure plots obtained with wavelet partitioning ('gaus1') and phase-space partitioning (*SFNN*). These two profiles of anomaly measure are generated from the same time series data, where the number of symbols $|\Sigma| = 8$ in both cases. It is observed that *ME* partitioning with 'gaus1' wavelet is comparable to *SFNN* partitioning. However, in this problem, the execution time for *SFNN* to generate the partition is found to be ≈ 4 h, while that for *ME* partitioning is ≈ 100 ms on the same computer. Therefore, it may be inferred from this experiment that *ME* partitioning is computationally several orders of magnitude less intensive than *SFNN* partitioning while they yield similar performance from the perspectives of anomaly detection.

5.2. Structural damage detection in mechanical systems

This example demonstrates efficacy of the *STSA* method for early detection of damage in mechanical structures. Laboratory experiments have been conducted on a multi-degree-of-freedom mechanical vibration apparatus [3]. The mechanical system in the apparatus is persistently excited at a frequency of 10.4 Hz, which is a close approximation of one of resonance frequencies of the mechanical structure. During the experiments, time series data have been collected from the measurements of displacement sensors, and each set contains 30 s of data under persistent vibratory motion of the mass-beam system. The resulting cyclic stresses induce (irreversible) fatigue crack damage in the critical structures, which cause gradual reduction in stiffness. Consequently, the statistics of time series data undergo changes. The objective here is to detect these changes as early as possible in real time.

The first data set, which is dominated by a sinusoid of frequency ~10.4 Hz and represents the nominal behavior of the mechanical vibration system, is considered to be the reference point. The wavelet 'gaus9' is chosen for analysis because this wavelet closely matches the shape of the signal. A small set of scales around the frequency of 10.4 Hz are chosen as per procedure outlined in Section 2. The alphabet size is chosen, based on the entropy rate, as eight, i.e., $|\Sigma| = 8$.

Fig. 12 depicts two plots of anomaly measure, which are obtained from the same data set with

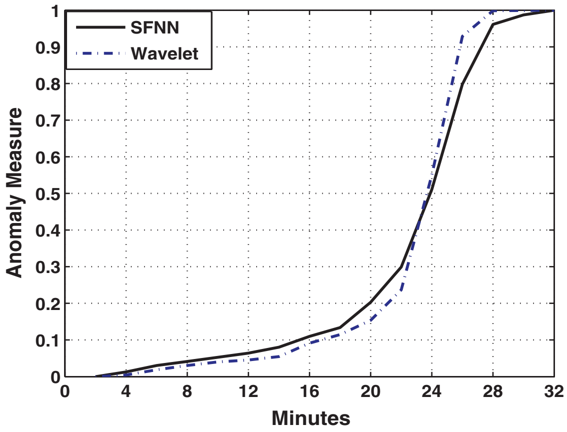


Fig. 12. Anomaly detection on the mechanical vibration apparatus.

different partitioning methods.

- *SFNN* partitioning [14].
- *ME* partitioning with wavelet 'gaus9'.

It is observed in Fig. 12 that *ME* partitioning with 'gaus9' wavelet is comparable to *SFNN* partitioning. However, in this problem, the execution time for *ME* partitioning is about five orders of magnitude less than that for *SFNN* partitioning. Hence, it may be inferred from this experiment that *ME* partitioning, is better suited for real-time structural damage detection in mechanical systems.

5.3. *Fatigue damage detection in polycrystalline alloys*

This example presents a comparison between wavelet space and phase-space partitioning in the context of fatigue crack damage detection in polycrystalline alloys. The details of the experimental apparatus, equipped with an ultrasonic flaw detector and an optical microscope, on which experiments were conducted are reported in [5] and references therein. Low-cycle fatigue tests have been conducted with specimens, made of aluminum alloy 7075-T6. A sinusoidal load with maximum and minimum loads of 87 and 4.85 MPa, respectively, is applied. A significant amount of internal damage occurs before the crack appears on the surface of the specimen, where it is observed by the optical microscope. This internal damage caused by multiple small cracks and dislocations affect the ultrasonic waves as they pass through the regions where these faults have developed. This phenomenon causes signal distortion and attenuation at the receiver end of the ultrasonic flaw detector.

The crack propagation stage starts when the internal damage eventually develops into a single large crack. Subsequently, the crack growth rate increases rapidly and when the crack is sufficiently large, the transmitted ultrasonic signal is almost completely attenuated. The process of obtaining the ultrasonic time series data is described in [5]. The results of *STSA*-based anomaly/damage detection, obtained with *ME* and *SFNN* partitioning methods, are presented below.

Wavelet 'gaus2' is used for obtaining the coefficients that are stacked to form the *scale series* data set. The alphabet size $|\Sigma|$ is chosen to be eight. The *scale series* data set is partitioned by imposing the ME criterion to generate the symbols. Fig. 13 depicts two anomaly measure plots that are obtained from the same data set by using *ME* and *SFNN* partitioning.

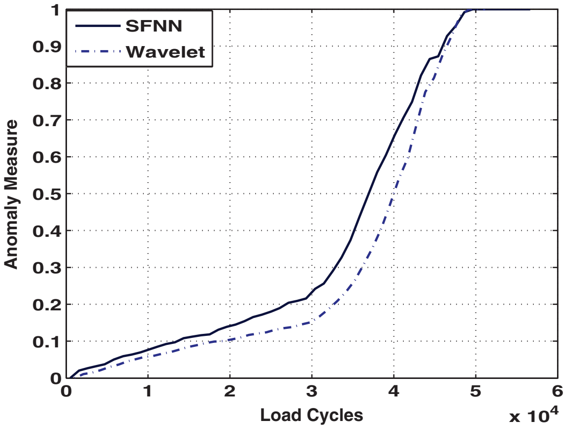


Fig. 13. Anomaly measure plots for fatigue crack detection.

While the *SFNN* partitioning yields slightly higher values of the anomaly measure and comparable slope, the profile of wavelet partitioning is more smooth and yields a larger change in the curvature around 32 kilocycles, which is an early warning for the forthcoming failure. Simultaneous consideration of the anomaly measure, slope, and curvature provides a robust method of failure prediction and reduces the probability of false alarms.

Similar to the previous experiments, it is observed that the execution time for *ME* partitioning is approximately five orders of magnitude less than that for *SFNN* partitioning. Hence, it may be inferred that *ME* partitioning, is better suited for real-time fatigue damage detection in polycrystalline alloys.

6. Summary and conclusions

This paper presents a novel method of partitioning the data space for symbolic time series analysis (*STSA*). In this approach, symbols are generated from the wavelet coefficients of the time series data, instead of obtaining them directly from the time series data. Various aspects of this method, such as selection of wavelet basis and scale range, are systematically investigated for: (i) suppression of noise and spurious disturbances and (ii) enhancement of sensitivity to changes in signal dynamics. The advantages of using wavelet-based partitioning over phase-space partitioning are demonstrated by numerical simulation and laboratory experimentation. It has been shown that measurement noise suppression results in robust symbol generation. The symbol sequences, generated from the wavelet coefficients of a noisy signal, are able to capture the changes in signal information better than those obtained directly from the time series data of the signal. It is also shown that the choice of an appropriate wavelet and scales significantly improves computational efficiency and thereby enhances implementation of the anomaly detection technique for real-time applications. An entropy-based algorithm is introduced for selection of the symbol alphabet size.

A partitioning method, based on maximum entropy, is introduced and is compared with the uniform partitioning method. The performance of wavelet-based partitioning has been tested via experimentation for: (i) anomaly detection on a nonlinear electronic system apparatus [1]; (ii) structural damage detection on a mechanical vibration system apparatus [3]; and (iii) damage detection in polycrystalline alloys on a Fatigue Testing apparatus [5].

Wavelet-based maximum entropy (*ME*) partitioning has been compared with symbolic false nearest neighbor (*SFNN*) partitioning [14] with regard to anomaly detection. Based on the time series data from three experimental apparatuses, it is observed that the aforementioned partitioning methods yield comparable results while the computation time for wavelet-based *ME* partitioning is observed to be several orders of magnitude smaller than that for *SFNN* partitioning.

A major conclusion based on this investigation is that wavelet-based *ME* partitioning, combined with an appropriate choice of wavelet and scales, significantly enhances computational efficiency and anomaly detection capabilities beyond what has been reported in literature. The field of STSA is relatively new and its application to anomaly detection is very recent. Therefore, the proposed method of symbol generation for anomaly detection requires continued theoretical and experimental research. In this context, future research is recommended in the following areas:

- Exploration of lifting techniques [23] for wavelet customization;
- Extension of *ME* partitioning to multi-dimensional time series;
- Noise reduction in time series for robust anomaly/damage detection.

Acknowledgements

The authors are grateful to Dr. Matthew Kennel for providing them with the software code on phase-space partitioning using symbolic false nearest neighbors. The authors acknowledge the expert technical advice of Dr. Eric Keller for design and fabrication of the experimental apparatuses. The authors also thank Mr. Amol Khatkhate and Mr. Shalabh Gupta for providing experimental data from the mechanical vibration apparatus and the fatigue testing apparatus.

References

[1] A. Ray, Symbolic dynamic analysis of complex systems for anomaly detection, Signal Processing 84 (7) (2004) 1115–1130.

[2] S. Chin, A. Ray, V. Rajagopalan, Symbolic time series analysis for anomaly detection: a comparative evaluation, Signal Processing 85 (9) (2005) 1859–1868.

[3] A. Khatkhate, A. Ray, S. Chin, V. Rajagopalan, E. Keller, Detection of fatigue crack anomaly: a symbolic dynamic approach, Proceedings of American Control Conference, Boston, MA, June–July 2004, pp. 3741–3746.

[4] D. Tolani, M. Yasar, A. Ray, V. Yang, Anomaly detection in aircraft gas turbine engines, AIAA J. Aerospace Comput., Inform. Comm. 3 (2006) 44–51.

[5] S. Gupta, A. Ray, E. Keller, Symbolic time series analysis of ultrasonic data for early detection of fatigue damage, in press, Elsevier. Corrected proof available online.

[6] S. Bhatnagar, V. Rajagopalan, A. Ray, Incipient fault detection in mechanical power transmission systems, Proceedings of American Control Conference, Portland, OR, 2005, pp. 472–477.

[7] R. Samsi, V. Rajagopalan, J. Mayer, A. Ray, Early detection of voltage imbalances in induction machines, Proceedings of American Control Conference, Portland, OR, June 2005, pp. 478–483.

[8] C.S. Daw, C.E.A. Finney, E.R. Tracy, A review of symbolic analysis of experimental data, Rev. Sci. Instrum. 74 (2) (2003) 915–930.

[9] H. Kantz, T. Schreiber, Nonlinear Time Series Analysis, second ed., Cambridge University Press, Cambridge, UK, 2004.

[10] C.J. Veenman, M.J.T. Reinders, E.M. Bolt, E. Baker, A maximum variance cluster algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 24 (9) (2002) 1273–1280.

[11] T. Chau, A.K.C. Wong, Pattern discovery by residual analysis and recursive partitioning, IEEE Trans. Knowledge Data Eng. 11 (6) (1999) 833–852.

[12] Y. Kakizawa, R.H. Shumway, N. Taniguchi, Discrimination and clustering for multivariate time series, J. Amer. Statist. Assoc. 93 (441) (1999) 328–340.

[13] T.W. Liao, Clustering of time series data—a survey, Pattern Recognition 38 (2005) 1857–1874.

[14] M.B. Kennel, M. Buhl, Estimating good discrete partitions form observed data: symbolic false nearest neighbors, Phys. Rev. E 91 (8) (2003) 084102.

[15] V. Rajagopalan, A. Ray, Wavelet-based space partitioning for symbolic time series analysis, Proceedings of IEEE Conference on Decision and Control (CDC) and European Control Conference (ECC), Seville, Spain, 2005, pp. 5245–5250.

[16] S. Mallat, A Wavelet Tour of Signal Processing, second ed., Academic Press, San Diego, CA, 1998.

[17] A. Teolis, Computational Signal Processing with Wavelets, Birkhäuser, Boston, MA, 1998.

[18] Wavelet Toolbox, MATLAB. Mathworks Inc, Natick, MA.

[19] P. Abry, Ondelettes et turbulence, multirésolutions, algorithmes de décomposition, invariance d'échelles, Diderot Editeur, Paris, 1997.

[20] G. Kaiser, A Friendly Guide to Wavelets, Birkhäuser, Boston, MA, 2004.

[21] J.M.T. Thompson, H.B. Stewart, Nonlinear Dynamics and Chaos, Wiley, Chichester, UK, 1986.

[22] V. Rajagopalan, R. Samsi, A. Ray, J. Mayer, C. Lagoa, A symbolic dynamics approach for early detection of slowly evolving faults in nonlinear systems, Proceedings of IASTED-CSS, Clearwater, FL, Paper No. 449–082, 2004.

[23] W. Sweldens, The lifting scheme: a construction of second generation wavelets, SIAM J. Math. Anal. 29 (2) (1998) 511–546.

Available online at www.sciencedirect.com
ScienceDirect

International Journal of Fatigue 29 (2007) 1100–1114

International Journal of Fatigue

www.elsevier.com/locate/ijfatigue

Online fatigue damage monitoring by ultrasonic measurements: A symbolic dynamics approach ☆

Shalabh Gupta, Asok Ray *, Eric Keller

Mechanical Engineering Department, The Pennsylvania State University, 329 Reber Building, University Park, PA 16802, United States

Received 1 December 2005; received in revised form 12 August 2006; accepted 10 September 2006
Available online 31 October 2006

Abstract

The paper presents an analytical tool for early detection and online monitoring of fatigue damage in polycrystalline alloys that are commonly used in mechanical structures of human-engineered complex systems. Real-time fatigue damage monitoring algorithms rely on time series analysis of ultrasonic signals that are sensitive to micro-structural changes occurring inside the material during the early stages of fatigue damage; the core concept of signal analysis is built upon the principles of *Symbolic Dynamics*, *Statistical Pattern Recognition* and *Information Theory*. The analytical tool of statistical pattern analysis has been experimentally validated on a special-purpose test apparatus that is equipped with ultrasonic flaw detection sensors and a travelling optical microscope. The paper reports fatigue damage monitoring of 7075-T6 alloy specimens, where the experiments have been conducted under load-controlled constant amplitude sinusoidal loadings for low-cycle and high-cycle fatigue.

Keywords: Symbolic time series analysis; Anomaly detection; Fatigue damage

1. Introduction

Prediction of structural damage and quantification of structural integrity are critical for safe and reliable operation of human-engineered complex systems. Fatigue damage is one of the most commonly encountered sources of structural degradation during both nominal and off-nominal operations of such systems [1]. Therefore, it is necessary to develop diagnosis and prognosis capabilities for reliable and safe operation of the system and for enhanced availability of its service life. Many model-based techniques have been reported in literature for structural health monitoring and life prediction [2–5]. Apparently, no existing model, solely based on the fundamental principles of molecular physics [6], can adequately capture the dynamical behavior of fatigue damage at the grain level. In general, model-based approaches are critically dependent on initial defects in the material micro-structure, which may randomly form crack nucleation sites and are difficult to model [1,7]. Small deviations in initial conditions and critical parameters may produce large bifurcations in the expected dynamical behavior of fatigue damage [8]. In addition, fluctuations in usage patterns (e.g., random overloads) and environmental conditions (e.g., temperature and humidity) may adversely affect the service life of mechanical systems. As such, fatigue damage is described as a stochastic phenomenon that emphasizes the need for online monitoring using sensing devices which can provide useful and reliable estimates of the anomalies at an early stage [9].

Information-based fatigue damage monitoring using different sensors (e.g., acoustic emission, eddy currents and ultrasonic) has been proposed in recent literature [10–12]. The capabilities of electrochemical sensors [13] and thermal imaging techniques [14] have also been investigated for structural failure analysis. The traditional analysis methods

☆ This work has been supported in part by the US Army Research Laboratory and the US Army Research Office (ARO) under Grant No. DAAD19-01-1-0646.

* Corresponding author.
E-mail addresses: szg107@psu.edu (S. Gupta), axr2@psu.edu (A. Ray), eek105@psu.edu (E. Keller).

0142-1123/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijfatigue.2006.09.011

Nomenclature

D	window length on a symbolic sequence
n	number of states of the finite state machine
$\mathbf{p}^k$	state probability vector at time epoch t_k
$P(\bullet)$	Probability of the event •
q_j	jth state of the finite state machine
Q	set of all states of the finite state machine
S	dynamical systems entropy of the symbol sequence
t_k	slow time epoch
σ	a symbol on a symbolic sequence
π^{jk}	transition probability from state q_j to q_k
Π^k	state transition matrix at time epoch t_k
ψ^k	anomaly measure computed at time epoch t_k
Φ_j	jth block of the partition of the phase space
Ω	phase space of a dynamical system
Σ	alphabet set
$\lvert\Sigma\rvert$	size of the alphabet set

using acoustic emission technique are used to correlate the signal parameters (such as the acoustic emission counts, the peak amplitudes and the energy) with the defect formation mechanisms and to provide a quantified estimate of faults. Acoustic emission technique has been investigated by several researchers for its sensitivity to the activities occurring inside the material micro-structure for early detection of fatigue and fracture failures [15–19]. However, the major drawback of acoustic emission technique is poor performance in noisy environments where signal–noise separation becomes a difficult task.

The eddy current technique is based on the principles of electromagnetic induction. When a source of alternating current is supplied to a conductor, a magnetic field develops, which induces eddy currents in the material. The presence of faults in the material affect the eddy current flow patterns, which can be detected for prediction and estimation of the structural damage [20–22]. The advantages of eddy current inspection technique include sensitivity to small cracks and other defects, portability of sensor equipment, minimum part preparation, and non-contact evaluation. However, the limitations of the eddy current inspection technique are the depth of penetration and it can be used to detect only surface and near surface defects. Furthermore, only conductive materials can be inspected.

Ultrasonic sensors function by emitting high-frequency ultrasonic pulses that travel through the specimen and are received by the transducers at the other end. As with the propagation of any wave, it is possible that discontinuities in the propagation media will cause additive and destructive interference. Since material characteristics (e.g., voids, dislocations and short cracks) influence the ultrasonic impedance, a small fault in the specimen is likely to change the signature of the signal at the receiver end [23,24,9,25,26]. Therefore, the ultrasonic signals can be used to capture some of the minute details and small changes during the early stages of fatigue damage, which may not be possible to detect by an optical microscope [9]. Moreover, ultrasonic sensing is applicable to real-time applications and the sensing probes can be easily installed at the critical sites. Ultrasonic sensing is also robust to noisy environments since the externally excited waves are of very high frequency and they do not interfere with small disturbances. As such, this paper explores the ultrasonic sensing technique to examine small micro-structural changes during early stages of fatigue damage evolution.

The above discussions evince the fact that time series analysis of sensor data is essential for real-time detection and monitoring of fatigue damage. From this perspective, the paper presents symbolic time series analysis (STSA) [27] of ultrasonic sensor signals for early detection of evolving anomalies. The STSA method is an information-theoretic pattern identification tool that is built upon a fixed-structure, fixed-order Markov chain [28]; it has been validated by comparison with existing pattern recognition techniques such as Principal Component Analysis (PCA) and Artificial Neural Networks (ANN) [29,30]. A computer-controlled special purpose test apparatus, equipped with multiple sensing devices (e.g., ultrasonics and optical microscope) for damage analysis, has been used to experimentally validate the STSA method of fatigue damage detection. Experiments have been conducted under different loading conditions on 7075-T6 aluminum alloy specimens.

The paper is organized in seven sections including the present section. Section 2 presents an overview of anomaly detection methodology using a two-time-scale approach. Section 3 describes the underlying concepts and essential features of symbolic time series analysis for anomaly detection. Section 4 provides a brief description of the fatigue damage test apparatus. Section 5 presents the experimental procedure and the application of STSA for fatigue damage monitoring. Section 6 presents the results and discussion on early detection of fatigue damage under different loading conditions. The paper is concluded in Section 7 along with recommendations for future research.

2. Methodology for fatigue damage monitoring

Traditional tools like *S*–*N* curves and Goodman diagrams are not effective for real-time damage prediction. This problem is circumvented by information-based diagnosis and prognosis tools that are capable of identifying damage patterns in real time from the statistical behavior of sensor (e.g., ultrasonic) data sequences. In this context,

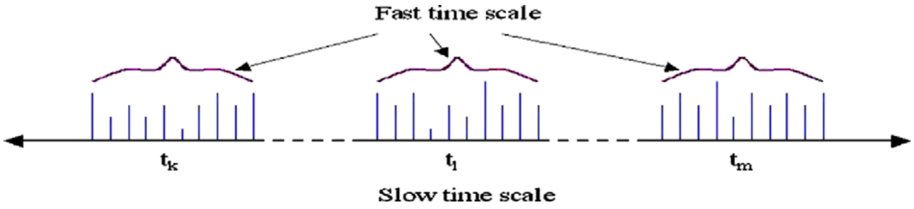


Fig. 1. Pictorial view of the two time scales: (1) *slow time scale* where anomalies evolve and (2) *fast time scale* where data acquisition is done.

fatigue damage monitoring is formulated as a two-time-scale problem. The sampling frequency for data acquisition is required to be several orders of magnitude faster than the time period of damage evolution. Fatigue damage monitoring is a two-time-scale problem as delineated below:

- The *fast time scale* is related to the response time of machinery operation. Over the span of a given time series data sequence, the structural dynamic behavioral statistics of the system are assumed to remain invariant, i.e., the process is assumed to have statistically stationary dynamics at the fast time scale.
- The *slow time scale* is related to the time span over which the process may exhibit non-stationary dynamics due to (possible) evolution of anomalies. Thus, an observable non-stationary behavior can be associated with anomalies evolving at a slow time scale.

A pictorial view of the two time scales is presented in Fig. 1. In general, a long time span in the fast time scale is a tiny (i.e., several orders of magnitude smaller) interval in the slow time scale. For example, evolution of fatigue damage in structural materials (causing a detectable change in the dynamics of the system) occurs on the slow time scale (possibly in the order of months or years); fatigue damage behavior is essentially invariant on the fast time scale (approximately in the order of seconds or minutes). Nevertheless, the notion of fast and slow time scales is dependent on the specific application, loading conditions and operating environment. As such, from the perspective of fatigue monitoring, sensor data acquisition is done on the fast time scale at different slow time epochs separated by fixed intervals on the slow time scale. Specifically, the objective of this paper is to demonstrate real-time monitoring of fatigue damage by detecting small changes in the statistical patterns of the ultrasonic sequences using the STSA approach.

3. Symbolic time series analysis for anomaly detection

This section presents the underlying concepts and essential features of symbolic time series analysis [27] for anomaly detection in complex dynamical systems [28]. While the details are reported in previous publications [28,29,31]; a brief review of this method is presented here for completeness and clarity of the paper. The principle of symbolic dynamics is based on the transformation of a data sequence (e.g., time series data) to a symbol sequence by partitioning a compact region Ω of the phase space, over which the trajectory evolves, into finitely many discrete blocks as shown in Fig. 2. Let $\{\Phi_1, \Phi_2, \ldots, \Phi_m\}$ be a partitioning of Ω, such that it is exhaustive and mutually exclusive set, i.e.,

$$\bigcup_{j=1}^{m} \Phi_j = \Omega \quad \text{and} \quad \Phi_j \bigcap \Phi_k = \emptyset \quad \forall j \neq k. \tag{1}$$

Each block Φ_j is labelled as the symbol $\sigma_j \epsilon \Sigma$, where the symbol set Σ is called the *alphabet set* consisting of m different symbols (i.e., $m = |\Sigma|$). As the system evolves in time, it travels through various blocks in its phase space and the corresponding symbol $\sigma_j \epsilon \Sigma$ is assigned to it, thus converting a data sequence to a symbol sequence $\ldots \sigma_{i_1} \sigma_{i_2} \ldots \sigma_{i_k} \ldots$ [32,33]. Fig. 2 exemplifies the partitioning of the phase space where each block is assigned a particular symbol such that a symbol sequence is generated from the phase space at a given slow time epoch. Once the symbol sequence is obtained, the next step is construction of a finite state machine [34]. These steps are explained in details in the following subsections.

3.1. Wavelet space partitioning

A crucial step in symbolic time series analysis is partitioning of the phase space for symbol sequence generation [27]. Several partitioning techniques have been reported in literature for symbol generation [35–37], primarily based on symbolic false neighbors. These techniques rely on partitioning the phase space and may become cumbersome and extremely computation-intensive if the dimension of the phase space is large. Moreover, if the time series data is noise-corrupted, then the symbolic false neighbors would rapidly grow in number and require a large symbol alphabet to capture the pertinent information on the system dynamics. Therefore, symbolic sequences as representations of the system dynamics should be generated by alternative methods because phase-space partitioning might prove to be a difficult task in the case of high-dimensions and presence of noise. The wavelet transform [38] largely alleviates these shortcomings and is particularly effective with noisy data from high-dimensional dynamical systems.

This paper has adopted a wavelet-based partitioning approach [28,31] for construction of symbol sequences from the time series data. In this method, the time series data are first converted by wavelet transform, where wavelet coefficients are generated at different scales and time shifts. The graphs of wavelet coefficients versus scale, at

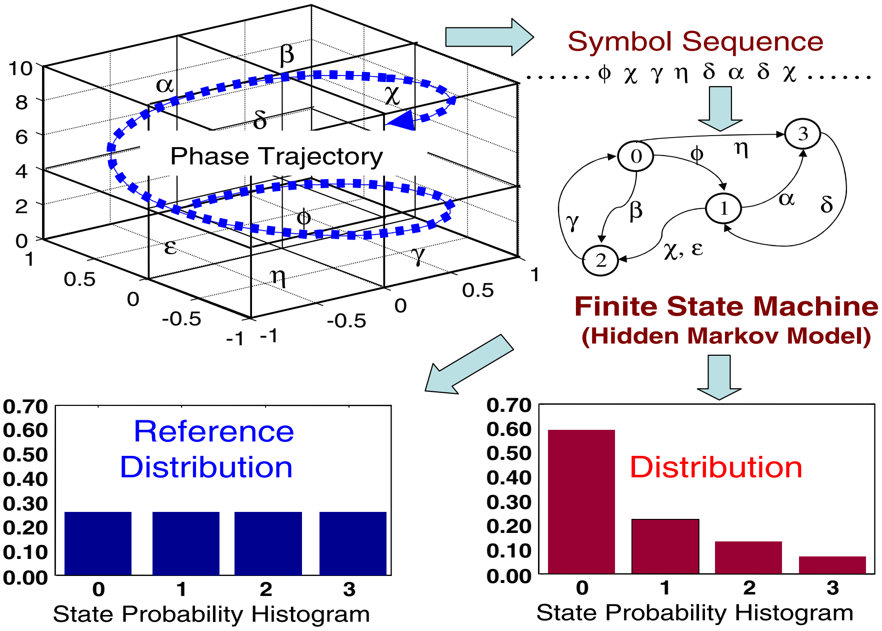


Fig. 2. STSA-based pattern identification.

selected time shifts, are stacked starting with the smallest value of scale and ending with its largest value and then back from the largest value to the smallest value of the scale at the next instant of time shift. The arrangement of the resulting *scale series* data in the wavelet space is similar to that of the time series data in the phase space. The wavelet space is partitioned with alphabet size $|\Sigma|$ into segments of coefficients on the ordinate separated by horizontal lines such that the regions with more information are partitioned finer and those with sparse information are partitioned coarser. This is achieved by maximizing the Shannon entropy [39], which is defined as

$$S = -\sum_{i=1}^{|\Sigma|} p_i \log(p_i), \tag{2}$$

where p_i is the probability of the ith state and summation is taken over all possible states. Uniform probability distribution of states is a consequence of maximum entropy that makes the partition coarser in regions of low-data density and finer in regions of high-data density.

3.2. State machine construction

The partitioning as described in the previous section is performed at time epoch t_0 of the nominal condition that is chosen to be a healthy condition having zero anomaly measure. A finite state machine [34] is then constructed, where the states of the machine are defined corresponding to a given *alphabet* Σ and window length D. The alphabet size $|\Sigma|$ is the total number of partitions while the window length D is the length of consecutive symbol words forming the states of the machine [28]. The states of the machine are chosen as all possible words of length D from the symbol sequence, thereby making the number n of states to be equal to the total permutations of the alphabet symbols within words of length D, (i.e., $n \leqslant |\Sigma|^D$; some of which may be forbidden with zero probability of occurrence). For example, if $\Sigma = \{0,1\}$, i.e., $|\Sigma| = 2$ and $D = 2$, then the number of states is $n \leqslant |\Sigma|^D = 4$; and the possible states are $\mathrm{Q} = \{00, 01, 10, 11\}$.

The choice of $|\Sigma|$ and D depends on specific experiments, noise level and also the available computation power. A large *alphabet* may be noise-sensitive while a small alphabet could miss the details of signal dynamics. Similarly, a high-value of D is extremely sensitive to small signal distortions but would lead to a large number of states requiring more computation power. Using the symbol sequence generated from the time series data, the state machine is constructed on the principle of sliding block codes [32] as explained below. The window of length D on the symbol sequence $\ldots\sigma_{i_1}\sigma_{i_2}\ldots\sigma_{i_k}\ldots$ is shifted to the right by one symbol, such that it retains the last $(D-1)$ symbols of the previous state and appends it with the new symbol σ_{i_ℓ} at the end. The symbolic permutation in the current window gives rise to a new state. The machine constructed in this fashion is called D-Markov machine [28] because of its Markov properties.

Definition 3.1. A symbolic stationary process is called D-Markov if the probability of the next symbol depends only on the previous D symbols, i.e., $P(\sigma_{i_0}/\sigma_{i_{-1}}\ldots\sigma_{i_{-D}}\sigma_{i_{-D-1}}\ldots) = P(\sigma_{i_0}/\sigma_{i_{-1}}\ldots\sigma_{i_{-D}})$.

The finite state machine constructed above has D-Markov properties because the probability of occurrence of symbol σ_{i_ℓ} on a particular state depends only on the configuration of that state, i.e., the previous D symbols. Once the partitioning *alphabet* Σ and word length D are determined at the nominal condition (time epoch t_0), they are kept constant for all (slow time) epochs $\{t_1, t_2, \ldots, t_k \ldots\}$, i.e., the structure of the machine is fixed at the nominal condition. That is, the partitioning and the state machine structure generated at the nominal condition serve as the reference frame for data analysis at subsequent time epochs. For $D = 1$, the set of states bears an equivalence relation to the alphabet Σ of symbols [40]. The states of the machine are marked with the corresponding symbolic word permutation and the edges joining the states indicate the occurrence of an event σ_{i_ℓ}. The occurrence of an event at a state may keep the machine in the same state or move it to a new state. The language of the machine is usually incomplete in the sense that all states might not be reachable from a given state.

Definition 3.2. The probability of transitions from state q_j to state q_k belonging to the set Q of states under a transition δ: $Q \times \Sigma \to Q$ is defined as

$$\pi_{jk} = P(\sigma \in \Sigma | \delta(q_j, \sigma) \to q_k); \sum_k \pi_{jk} = 1. \tag{3}$$

Thus, for a D-Markov machine, the irreducible stochastic matrix $\Pi \equiv [\pi_{ij}]$ describes all transition probabilities between states such that it has at most $|\Sigma|^{D+1}$ nonzero entries. The left eigenvector $\mathbf{p}$ corresponding to the unit eigenvalue of Π is the state probability vector under the (fast time scale) stationary condition of the dynamical system [28]. On a given symbol sequence $\ldots \sigma_{i_1} \sigma_{i_2} \ldots \sigma_{i_l} \ldots$ generated from the time series data collected at slow time epoch t_k, a window of length (D) is moved by keeping a count of occurrences of word sequences $\sigma_{i_1} \ldots \sigma_{i_D} \sigma_{i_{D+1}}$ and $\sigma_{i_1} \ldots \sigma_{i_D}$ which are respectively denoted by $N(\sigma_{i_1} \ldots \sigma_{i_D} \sigma_{i_{D+1}})$ and $N(\sigma_{i_1} \ldots \sigma_{i_D})$. Note that if $N(\sigma_{i_1} \ldots \sigma_{i_D}) = 0$, then the state $q \equiv \sigma_{i_1} \ldots \sigma_{i_D} \in Q$ has zero probability of occurrence. For $N(\sigma_{i_1} \ldots \sigma_{i_D}) \neq 0$, the transitions probabilities are then obtained by these frequency counts as follows:

$$\pi_{jk} \equiv P[q_k | q_j] = \frac{P[q_k, q_j]}{P[q_j]} = \frac{P(\sigma_{i_1} \ldots \sigma_{i_D} \sigma)}{P(\sigma_{i_1} \ldots \sigma_{i_D})} \Rightarrow \pi_{jk} \approx \frac{N(\sigma_{i_1} \ldots \sigma_{i_D} \sigma)}{N(\sigma_{i_1} \ldots \sigma_{i_D})}, \tag{4}$$

where the corresponding states are denoted by $q_j \equiv \sigma_{i_1} \sigma_{i_2} \ldots \sigma_{i_D}$ and $q_k \equiv \sigma_{i_2} \ldots \sigma_{i_D} \sigma$. The time series data under the nominal condition (set as a benchmark) generates the *state transition matrix* Π^0 that, in turn, is used to obtain the *state probability vector* $\mathbf{p}^0$ whose elements are the stationary probabilities of the state vector, where $\mathbf{p}^0$ is the left eigenvector of Π^0 corresponding to the (unique) unit eigenvalue. Subsequently, state probability vectors $\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^k, \ldots$ are obtained at slow time epochs $t_1, t_2, \ldots, t_k, \ldots$ based on the respective time series data. Machine structure and partitioning should be the same at all slow time epochs.

3.3. Pattern identification procedure

Behavioral pattern changes are quantified as deviations from the nominal behavior (i.e., the probability distribution at the nominal condition). The resulting anomalies (i.e., deviations of the evolving patterns from the nominal pattern) are characterized by a scalar-valued function, called *Anomaly Measure* ψ that is quasi-static in the fast time scale and is monotonically non-decreasing in the slow time scale. The state probability vector at any time instant corresponds to a singleton point on the unity-radius hypersphere. During fatigue damage evolution, the tip of the probability vector moves along a path on the surface of this hypersphere. The initial starting point of the path is the probability vector with uniform distribution obtained with maximum entropy partitioning (see Section 3.1). As the damage progresses, the probability distribution changes; eventually when a very large crack is formed, complete attenuation of the ultrasonic signal occurs and consequently the tip of the probability vector reaches a point where all states have zero probabilities of occurrence except one which has a probability one (i.e., a delta-distribution); this state corresponds to the partition region where all data points are clustered due to complete attenuation of the signal.

In the context of fatigue damage, the anomaly measure is formulated on the following assumptions.

- *Assumption* #1: The damage evolution is an irreversible process (i.e., with zero probability of self healing) and implies the following conditions.

$$\psi^k \geqslant 0; \ \psi^{k+\ell} - \psi^k \geqslant 0 \quad \forall \ell \geqslant 0 \quad \forall k. \tag{5}$$

- *Assumption* #2: The damage accumulation between two time epochs is a path function, i.e., dependent on the path traversed to reach the target state from the initial state.

In the context of fatigue damage in polycrystalline alloys at room temperature, the crack length is traditionally defined by a straight line joining the starting point to the tip of the crack but, in reality, the actual crack follows a complicated path (possibly fractal in ductile materials). In fact, at the initial stages of fatigue damage, there can be multiple short cracks oriented in different directions. Therefore, crack length alone does not provide complete information on fatigue damage evolution. Since ultrasonic signals are highly sensitive to small micro-structural changes, signal distortion is a good index of anomaly growth. The tip of the probability vector, obtained through symbolic time series analysis, moves along a curved path on the surface of the unity-radius hypersphere between the initial point $\mathbf{p}^0$ (i.e., uniform distribution obtained under maximum entropy partitioning) and the final point at very large crack formation $\mathbf{p}^f$ (i.e., δ-distribution due to complete attenuation of the signal). The phenomenon such as piling up of dislocations, strain hardening or reflections from multiple crack surfaces affect the ultrasonic signals in a variety of ways. An increase of the ultrasonic amplitude is also observed during very early stages of fatigue damage due to hardening of the material. On the other hand, ultrasonic signals attenuate sharply at the crack propagation stage upon development of a large crack.

As such, distortion of ultrasonic signals at a single time epoch may not uniquely determine the state of fatigue damage. The rationale is that two signals may exhibit similar characteristics but, in terms of actual incurred damage, the states are entirely different. Consequently, fatigue damage is a path function instead of being a state function. This assessment is consistent with assumption #1 implying that the damage evolution is irreversible. That is, at two different time epochs, the damage cannot be identical unless the net damage increment is zero. Consequently, by assumption #2, the anomaly measure should follow the traversed path of the probability vector, not the straight line joining the end points (i.e., the tips of the probability vectors).

The anomaly measure, based on the path between the nominal state to the completely damaged state, can be different even for identical test samples and under the same loading conditions because of the stochastic nature of fatigue phenomena. As such, analysis of a stochastic data set collected under identical experimental conditions is essential for identification of variations in different data sets. The following distance function is derived between probability vectors at two time epochs:

$$d(\mathbf{p}^k,\mathbf{p}^l) \equiv \sqrt{(\mathbf{p}^k-\mathbf{p}^l)^{\mathrm{T}}(\mathbf{p}^k-\mathbf{p}^l)}. \tag{6}$$

The algorithm for computation of the anomaly measure ψ compensates for spurious measurement and computation noise in terms of the sup norm which is defined as $\|\mathbf{e}\|_\infty \equiv \max(|e_1|,\ldots,|e_m|)$ of the error in the probability vector (i.e., the maximum error in the elements of the probability vector). The algorithm is presented below.

```
(i) $\psi^0 = 0$; $\delta\psi^1 = 0$; $\widetilde{\mathbf{p}} = \mathbf{p}^0$; $k = 1$;
(ii) if $\|\mathbf{p}^k - \widetilde{\mathbf{p}}\|_\infty > \epsilon$ then $\delta\psi^k = d(\mathbf{p}^k,\widetilde{\mathbf{p}})$ and $\widetilde{\mathbf{p}} \leftarrow \mathbf{p}^k$;
(iii) $\psi^k = \psi^{k-1} + \delta\psi^k$;
(iv) $k \leftarrow k+1$; $\delta\psi^k = 0$; go to step (ii).
```

The real positive parameter ϵ, is associated with the robustness of the measure against measurement and computation noise and is identified by performing an experiment with a sample with no notch. Since there is no notch there is practically no stress augmentation and relatively no fatigue damage. As such, the parameter ϵ is estimated as

$$\epsilon \approx \max_{l\in\{1,\ldots,N\}}(\|\mathbf{p}^{l+1}-\mathbf{p}^l\|_\infty) \tag{7}$$

from N consecutive observations with $N \gg 1$.

The algorithm works in the following fashion: the reference point $\widetilde{\mathbf{p}}$ is initialized to the starting point $\mathbf{p}^0$ and anomaly measure ψ^0 is set to 0. At any slow time epoch t_k if the state probability vector moves such that the distance travelled in any particular direction (i.e., the sup norm $\|\bullet\|_\infty$) is greater than ϵ as specified in step (ii), then the anomaly measure is incremented by $\delta\psi^k = d(\mathbf{p}^k,\widetilde{\mathbf{p}})$ and the reference point is shifted to the current point $\mathbf{p}^k$. The procedure is repeated at all slow time epochs. As such, the total path travelled by the tip of probability vector represents the deviation from the nominal condition and the associated damage.

3.4. Real-time implementation

Fatigue damage monitoring using STSA has been successfully implemented in real time. The nominal condition is chosen after the start of the experiment at time epoch t_0, when the system attains a steady state and is considered to be in a healthy condition. The function module for STSA is triggered at this point. The D-Markov machine states are fixed in advance using *a priori* determined values of the parameters: alphabet size $|\Sigma|$ and window length D. The tasks of wavelet space partitioning and D-Markov machine construction are performed based on the time series data at the slow time epoch t_0 (nominal condition). The state probability vector $\mathbf{p}^0$ at time epoch t_0 is stored for computation of anomaly measures at subsequent slow time epochs, $t_1, t_2, \ldots, t_k, \ldots$, which are chosen to be separated by uniform intervals of time in these experiments. The ultrasonic data files at time epochs $t_0, t_1, \ldots, t_k, \ldots$ are read by the STSA function module that calculates the anomaly measure values at these time epochs. The algorithm is computationally very fast (i.e., several orders of magnitude faster relative to slow time scale damage monitoring) and the evolution of anomaly measure is exhibited in real time. The plot is updated with the most recent value of anomaly measure at each (slow time) epoch. Thus, the STSA algorithm allows on-line health monitoring and is capable of issuing warnings of incipient failures well in advance.

3.5. Advantages of STSA

After having discussed the underlying principles and essential features of STSA, the major advantages of STSA for anomaly detection are listed below:

(a) *Robustness to measurement noise and spurious signals* [31] – The procedure of STSA is robust to measurement noise and spurious disturbances and it filters out the noise at different steps. First of all, coarse graining of the continuous data (i.e., partitioning into finite blocks) and generation of a symbol sequence eliminate small measurement noise [28]. Secondly, the wavelet transform also contributes in signal–noise separation of the raw time series data by proper choice of scales [31]. Finally, the state probabilities are generated by passing a long symbol sequence over the finite state machine, which further eliminates small (zero-mean) measurement noise.

(b) Adaptability to low-resolution sensing due to coarse graining in space partitions [28].
(c) Capability for early detection of anomalies because of sensitivity to signal distortion and real-time execution on commercially available inexpensive platforms [30,29].
(d) Applicability to networked communication systems due to the capability of data compression into low-dimensional pattern vectors.

3.6. Summary of STSA based anomaly detection

The symbolic time series analysis (STSA) procedure of anomaly detection is summarized below.

- Acquisition of time series data from appropriate sensor(s) at time epoch t_0 of the nominal condition, when the system is assumed to be in the healthy state (i.e., zero anomaly measure).
- Generation of the wavelet transform coefficients, obtained with an appropriate choice of the wavelet basis and scale range.
- Maximum entropy partitioning of the wavelet *scale series data* at the nominal condition (see Section 3.1); and generation of the corresponding symbol sequence. The partitioning is fixed for subsequent time epochs.
- Construction of the D-Markov machine states from the symbol alphabet size $|\Sigma|$ and the window length D, and generation of the state probability vector $\mathbf{p}^0$ at time epoch t_0.
- Generation of time series data sequences at subsequent slow time epochs, $t_1, t_2, \ldots, t_k, \ldots$, and their conversion to the wavelet domain to generate respective symbolic sequences based on the partitioning at time epoch t_0.
- Generation of the state probability vectors $\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^k, \ldots$ at slow time epochs, $t_1, t_2, \ldots, t_k, \ldots$ from the respective symbolic sequences using the finite state machine constructed at time epoch t_0.
- Computation of scalar *Anomaly Measures* $\psi^1, \psi^2, \ldots, \psi^k, \ldots$ at time epochs, $t_1, t_2, \ldots, t_k, \ldots$

4. Experimental apparatus and sensors for damage detection

The experimental apparatus, shown in Fig. 3, is a special-purpose uniaxial fatigue testing machine, which is operated under load control or strain control at speeds up to 12.5 Hz; a detailed description of the apparatus and its design specifications are reported in [41]. The test specimens are subjected to tensile–tensile cyclic loading by a hydraulic cylinder under the regulation of computer-controlled electro-hydraulic servo-valves. The damage estimation and life prediction subsystem consists of data analysis software and the associated computer hardware. The process instrumentation and the control module of the fatigue test apparatus are briefly described below.

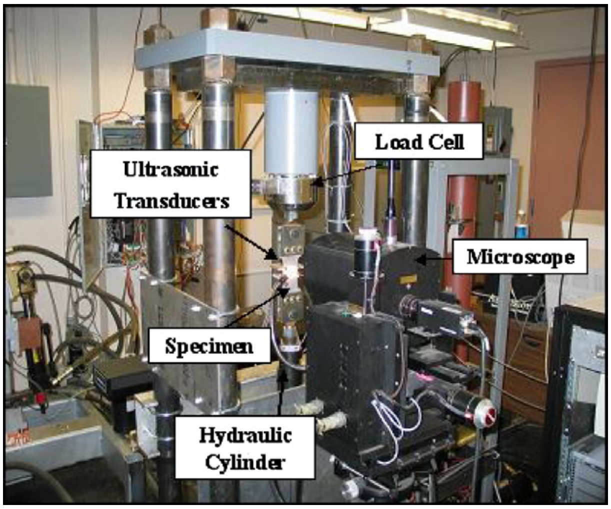


Fig. 3. Computer-instrumented apparatus for fatigue testing.

- *Closed loop servo-hydraulic unit and controller*: The instrumentation and control of the computer-controlled uniaxial fatigue test apparatus includes a load cell, an actuator, the hydraulic system, and the controller. The servo-hydraulic unit can provide either random loads or random strains to a specimen for both low-cycle and high-cycle fatigue tests at variable amplitudes and multiple frequencies. The control module is installed on a computer which is dedicated to machine operation. The controller runs the machine according to a schedule file which contains the loading profile and the number of load cycles. The real time data from the extensometer and load cell are supplied to the controller for operation under specified position and load limits.
- *Subsystem for data acquisition, signal processing, and engineering analysis*: In addition to the computer for controlling the load frame, a second computer is used for real-time image data collection from the microscope to monitor the growth of surface cracks. This computer controls the movement of the microscope to focus on the region of the crack tip. The instrumentation for ultrasonic flaw detection scheme is connected to a third computer. The ultrasonic data collected on this computer in real-time is then transferred at regular intervals to a fourth computer on which the STSA software is installed. This computer performs the real-time data analysis task. These laboratory computers are interconnected by a local dedicated network for data acquisition, data communications, and control.

Fig. 4 shows a typical 7075-T6 aluminum specimen used for testing in the fatigue damage test apparatus. The specimens used are 3 mm thick and 50 mm wide, and have a slot of 1.58 mm × 4.5 mm at the center. The central notch is made to increase the stress concentration factor that ensures crack initiation and propagation at the notch ends. The test specimens have been subjected to sinusoidal loading under tension–tension mode (i.e., with a constant posi-

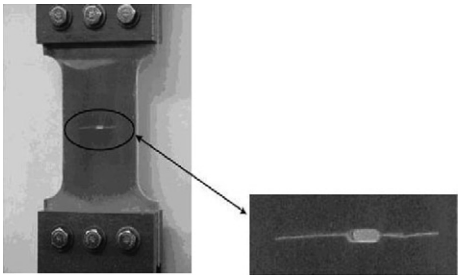


Fig. 4. Cracked specimen with a central notch.

tive offset) at a frequency of 12.5 Hz. The DC offset was provided in the load cycling to ensure that the specimen was always under tension. Since inclusions and flaws are randomly distributed across the material, small cracks appear at these defects and propagate and join at the machined surface of the notch even before microscopically visual cracks appear on the surface.

The test apparatus is equipped with two types of sensors that have been primarily used for damage detection:

(1) *Travelling optical microscope*: The travelling optical microscope, shown as part of the test apparatus in Fig. 3, provides direct measurements of the visible portion of a crack. The resolution of the optical microscope is about 2 μm at a working distance of 10–35 cm and the images are taken at a magnification of 75×. The growth of the crack is monitored continuously by the microscope which takes the images of the surface of the specimen at regular intervals. The microscope shifts from left to right side of the central notch and vice versa after every 200 cycles to track crack growth on both sides of the notch. In order to take pictures the controller slows down the machine to less than 5 Hz to get a better resolution of the images. The crack length can be calculated automatically by movement of the microscope from the respective notch end to the tip of the crack. The data acquisition software also allows for manual operation and image capture at the desired moment. Formation of very small cracks is difficult to detect and model due to large variability of material irregularities. This paper primarily focuses on analyzing ultrasonic data for more accurate characterization of the nature of small defects.

(2) *Ultrasonic flaw detector*: A piezoelectric transducer is used to inject ultrasonic waves in the specimen and an array of receiver transducers is placed on the other side of notch to measure the transmitted signal. In these experiments, an array of 2 receiver transducers was placed below the notch to detect faults on both left and right side of the notch. The ultrasonic waves produced were 5 MHz sine wave signals and they were emitted during a very short portion at the peak of every load cycle. Ultrasonic measurements were taken at stress levels that exceeded the crack opening stress and this causes maximum attenuation of the ultrasonic waves. Note that if crack closure occurs at low-loads, then an alternative method would be needed to detect anomalies. The sender and receiver ultrasonic transducers are placed on two positions, above and below the notch, so as to send the signal through the region of crack propagation and receive it on the other side, as seen in Fig. 5.

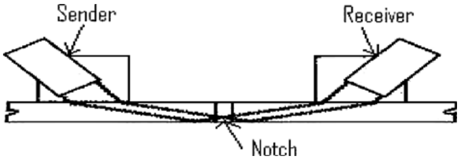


Fig. 5. Schematic of ultrasonic sensors on a test specimen.

As with the propagation of any wave, it is possible that discontinuities in the propagation media will cause additive and destructive interference. Since material characteristics (e.g., voids, dislocations and short cracks) influence ultrasonic impedance, a small fault in the specimen is likely to change the signature of the signal at the receiver end. The effect of these discontinuities in the material is to distort the transmitting ultrasonic waves. Since ultrasonic waves have a very small wavelength, very small faults can be detected. Therefore, the received signal can be used to capture minute details and small changes during the early stages of fatigue damage, which are not possible to detect by an optical microscope [9]. Prior to the appearance of a single large crack on the surface of the specimen as detected by the optical microscope, deformations (e.g., dislocations and short cracks) can cause detectable attenuation and/or distortion of the ultrasonic waves [25]. Recent literature has also shown nonlinear modelling approaches of the ultrasonic interference with the material micro-structures [42,43]. An elaborate description of the properties of ultrasonic waves in solid media is provided by Rose [44].

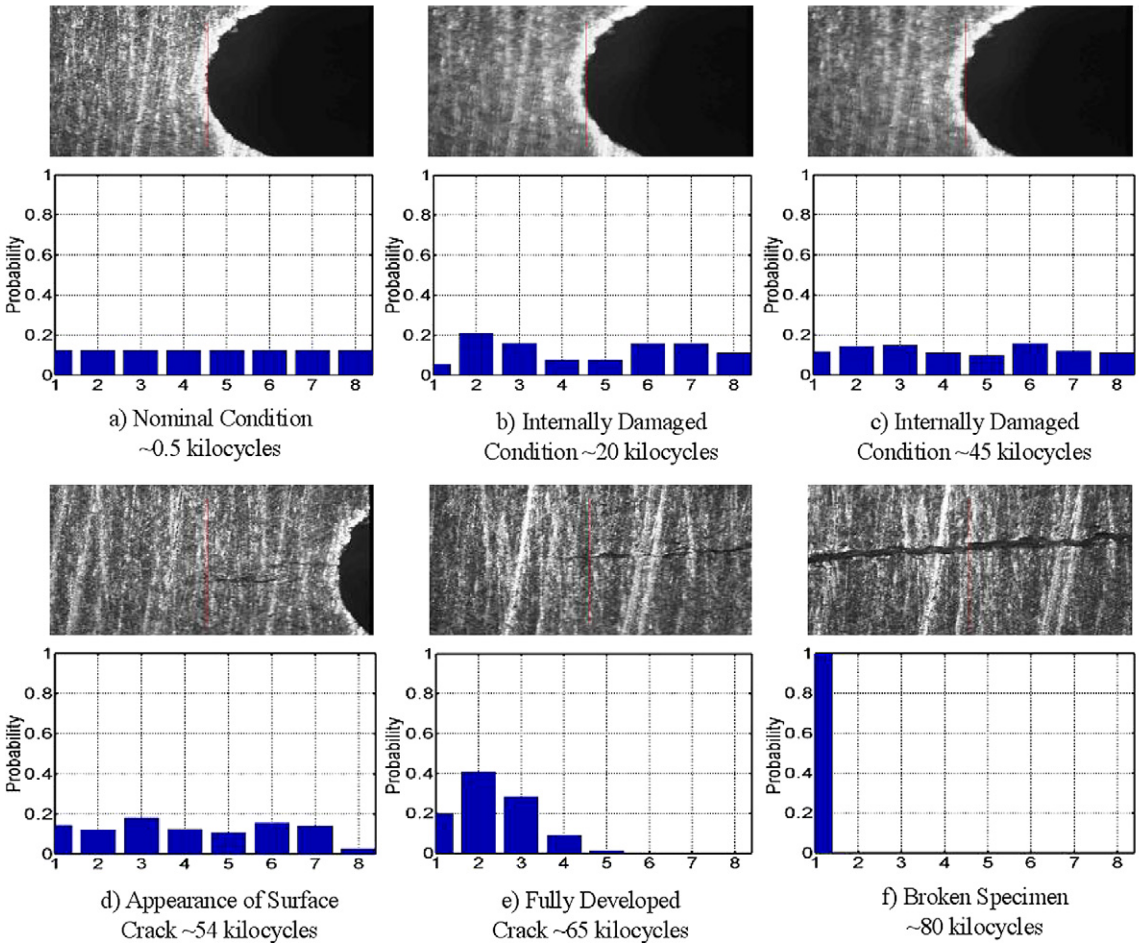
The advantages of using ultrasonic flaw detection over a microscope are the ease of installation at the desired damage site and detection of early anomalies before the onset of widespread fatigue crack propagation. It is observed that a crack always starts at the stress-concentrated region near the notch but the exact site of crack nucleation can be treated as a random event. An optical microscope is only capable of detecting cracks when they appear on the front surface of the specimen. Therefore, the study in this paper is based on analyzing the ultrasonic data for identification of fatigue damage in the small crack regime.

5. Experimentation and application of *STSA* for fatigue damage monitoring

The fatigue tests were conducted on center notched specimens with the objective of real-time continuous monitoring of fatigue damage growth on both sides of the notch. The tests were performed at 12.5 Hz frequency under two different types of loading conditions: (a) *stress controlled low-cycle fatigue* and (b) *stress controlled high-cycle fatigue*. For low-cycle fatigue loading the specimens were subjected to a sinusoidal load where the maximum and minimum loads were kept constant at 92.5 MPa and 4.85 MPa. For high-cycle fatigue loading the maximum and minimum amplitudes were kept constant at 71 MPa and 4.85 MPa.

The optical images were collected automatically at every 200 cycles until a crack was detected on the specimen surface by the optical microscope. Subsequently, the images were taken at user command and the microscope was moved such that it always focused on the crack tip. A significant amount of micro-structural damage caused by multiple small cracks, dislocations and other defects occurs before a single large crack appears on the surface of the specimen when it is observed by the optical microscope [45]. This phenomenon causes distortion and attenuation of the ultrasonic signal at the receiver end. The crack propagation stage starts when this micro-structural damage eventually develops into a single large crack. Subsequently, the crack growth rate increases rapidly and when the crack is sufficiently large, complete attenuation of the transmitted ultrasonic signal occurs, as seen at the receiver end. After the crack appears on the surface, fatigue damage growth can be easily monitored by the microscope but the ultrasonics provide early warnings even during the crack initiation phase.

Ultrasonic waves with a frequency of 5 MHz were triggered at each peak of the sinusoidal load to obtain 100 data points in each cycle. Since the ultrasonic frequency is much higher than the load frequency, data acquisition was done for a very short interval in the time scale of load cycling. Therefore, it can be implied that ultrasonic data were collected at the peak of each sinusoidal load cycle, where the stress is maximum and the crack is open causing maximum attenuation of the ultrasonic waves. The slow time epochs were chosen to be 1000 load cycles (i.e., ∼80 s) apart. At the onset of each slow time epoch, the ultrasonic data points were collected on the fast time scale of 100 cycles (i.e., ∼8 s), which produced a string of 10,000 data points. It is assumed that during this fast time scale of 100 cycles, the system remained in a stationary condition and no major changes occurred in the fatigue damage behavior. This set of time series data collected in the manner described above at different slow time epochs was analyzed using the STSA method to calculate the anomaly measures at respective slow time epochs.



a) Nominal Condition ~0.5 kilocycles

b) Internally Damaged Condition ~20 kilocycles

c) Internally Damaged Condition ~45 kilocycles

d) Appearance of Surface Crack ~54 kilocycles

e) Fully Developed Crack ~65 kilocycles

f) Broken Specimen ~80 kilocycles

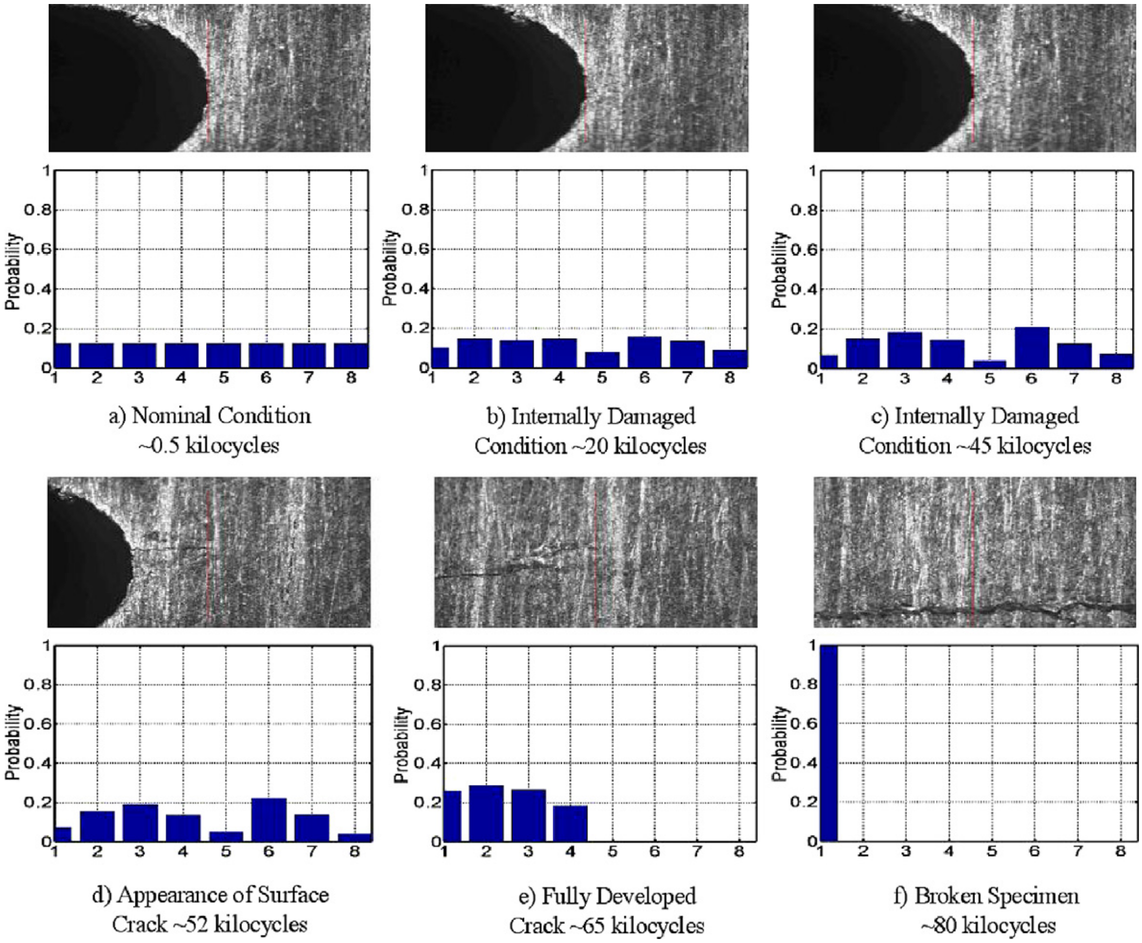
Fig. 6. Evolution of surface crack and probability distribution on left side of the notch under low cycle fatigue.

Following the STSA procedure for anomaly detection as described in Section 3, the alphabet size and depth have been chosen to be $|\Sigma| = 8$ and $D = 1$, respectively; consequently, the number of machine states is $|\Sigma|^D = 8$. The wavelet basis has been chosen to be ‘gaus2’ [46]. Absolute values of the wavelet scale series data have been used to generate the partition because of the symmetry of the data sets about their mean. With this selection, the STSA tool has been able to capture the anomalies significantly earlier than the optical microscope. Increasing the value of $|\Sigma|$ further did not improve the results and increasing the value of D created a large number of states of the finite state machine, many of them having very small or zero probabilities, and required a larger number of data points at each time epoch to stabilize the state probability vectors. The advantage of having a small number of states is fast computation on inexpensive processors and also robustness to noise. The wavelet basis, ‘gaus2’, provides better results than the wavelet bases of the Daubechies family [38] because the ‘gaus2’ wavelet base closely matches the shape of the sinusoidal ultrasonic signals.

The nominal condition at the slow time epoch t_0 was chosen to be $\sim$0.5 kilocycles to ensure that the electro-hydraulic system of the test apparatus had come to a steady state and that no significant damage occurred till that point. This nominal condition was chosen as a benchmark where the specimen was assumed to be in a healthy state, and thus the anomaly measure was chosen to be zero. The anomalies at subsequent slow time epochs, $t_1, t_2, \ldots, t_k, \ldots$, were then calculated with respect to the nominal condition at t_0. It is emphasized that the anomaly measure is relative to the nominal condition which is fixed in advance and should not be confused with the actual damage at an absolute level. However, inferring fatigue damage from the observed anomaly measure is an inverse problem that is a topic of future research.

6. Experimental results and discussion

This section presents the results of symbolic time series analysis of the ultrasonic data generated by fatigue tests described in Section 5. In each of the six plate pairs from (a) to (f) in Figs. 6–9, the top plate displays the surface image of the test specimen, as seen by the optical microscope; and histograms of the state probability distribution in the bottom plates exhibit evolution of fatigue damage patterns at different slow time epochs. These patterns gradually change from uniform distribution (i.e., minimal



a) Nominal Condition ~0.5 kilocycles

b) Internally Damaged Condition ~20 kilocycles

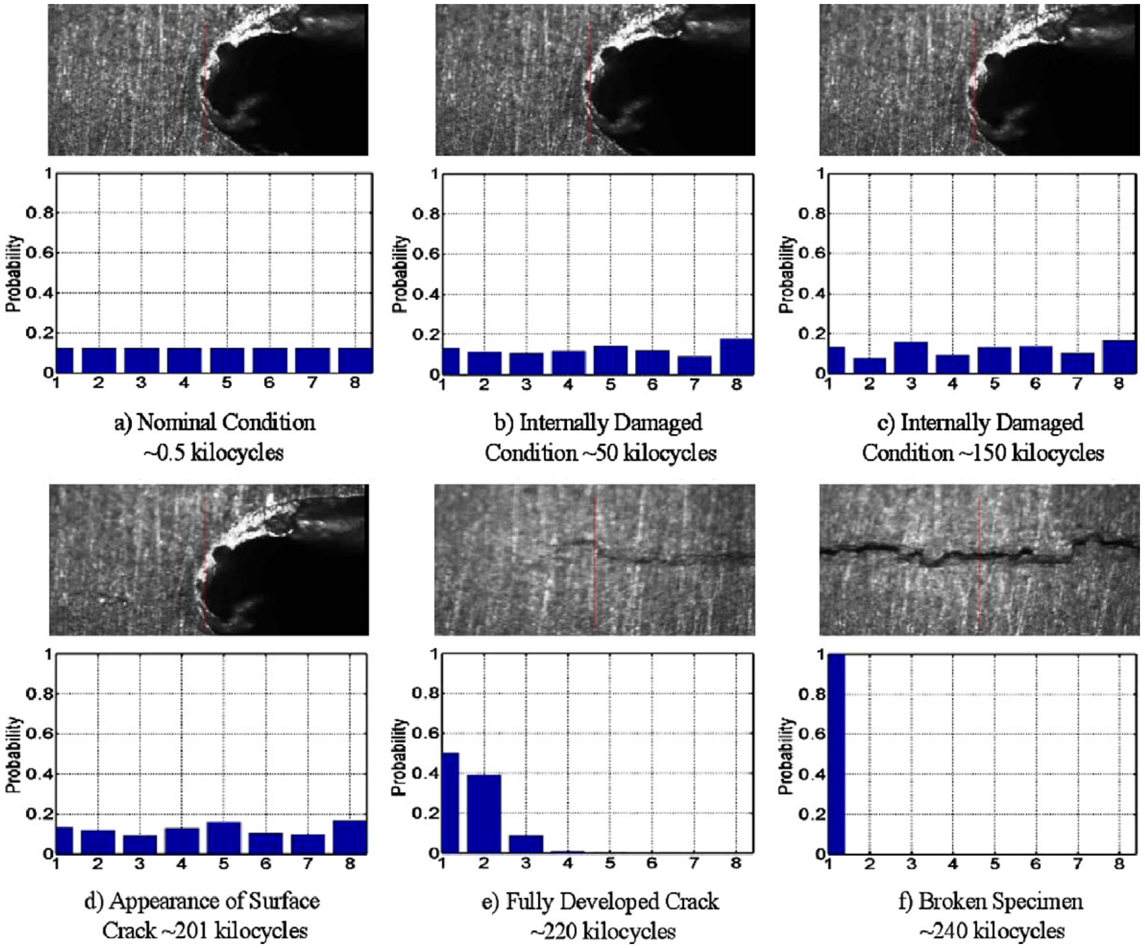
c) Internally Damaged Condition ~45 kilocycles

d) Appearance of Surface Crack ~52 kilocycles

e) Fully Developed Crack ~65 kilocycles

f) Broken Specimen ~80 kilocycles

Fig. 7. Evolution of surface crack and probability distribution on right side of the notch under low cycle fatigue.



a) Nominal Condition ~0.5 kilocycles

b) Internally Damaged Condition ~50 kilocycles

c) Internally Damaged Condition ~150 kilocycles

d) Appearance of Surface Crack ~201 kilocycles

e) Fully Developed Crack ~220 kilocycles

f) Broken Specimen ~240 kilocycles

Fig. 8. Evolution of surface crack and probability distribution on left side of the notch under high-cycle fatigue.

information) to delta distribution (i.e., maximum information) in each figure.

The image in each of Figs. 6(a), 7(a), 8(a) and 9(a) shows the nominal condition at ∼0.5 kilocycles when the anomaly measure is taken to be zero. This is considered to be the reference point with the available information on potential damage being minimal. This is reflected in the uniform distribution (i.e., maximum entropy or highest uncertainty) as seen from the histograms in the corresponding bottom plates.

The images in plate pairs (b) and (c) in Figs. 6–9 show the specimen surface before crack propagation. These images do not yet have any indication of surface crack although the corresponding bottom plates do exhibit deviations from the uniform probability distribution (see the bottom plate in plate pair (a) for comparison). This is an evidence that the analytically derived results, based on ultrasonic sensor data, produce relevant damage information during crack initiation; this information is not available from the corresponding optical images. For low-cycle fatigue in Figs. 6 and 7, the plate pairs (b) and (c) represent damage status at ∼20 and ∼45 kilocycles, respectively; and for high-cycle fatigue in Figs. 8 and 9, they represent damage status at ∼50 and ∼150 kilocycles, respectively. The deviations of these histograms from those at the nominal condition in plates (a) indicate that the probability vector has moved to another location on the surface of unity-radius hypersphere. The path depends on the interactions of the ultrasonic signals with the subsurface deformities.

The image in each of Figs. 6(d), 7(d), 8(d) and 9(d) exhibits the first noticeable appearance of a crack on the specimen surface, which is often deemed as the boundary of crack initiation and crack propagation phases; however, crack propagation might have started before the appearance of a surface crack. For low-cycle fatigue in Figs. 6 and 7, surface cracks have been detected by the microscope at ∼54 and ∼52 kilocycles on the left side and right side of the notch, respectively. Similarly, for high-cycle fatigue in Figs. 8 and 9, surface cracks have been detected at ∼201 and ∼215 kilocycles on left side and right side of the notch, respectively. The appearance of a large surface crack indicates that a significant portion of the crack or multiple small cracks might have already developed underneath the surface before they started spreading on the surface. However, further micro-structural analysis is need to confirm these findings. The histogram of probability distribution in the corresponding bottom plates show further

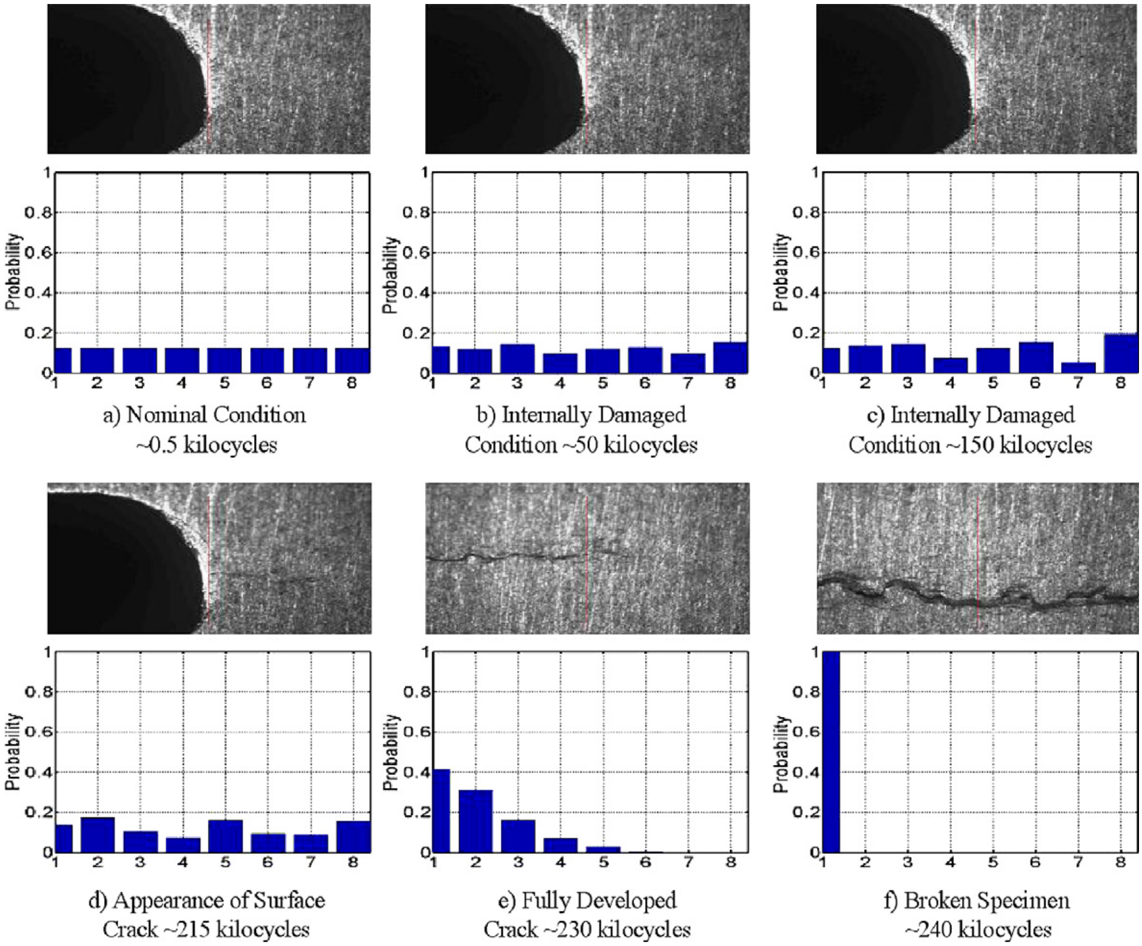


Fig. 9. Evolution of surface crack and probability distribution on right side of the notch under high-cycle fatigue.

evolution of the probability vector with the fatigue damage. At this stage, the information on damage is enhanced compared to what was available in the earlier cycles. At the onset of the crack propagation phase, the histograms of probability distribution show drastic changes in patterns, which indicate rapid development of large cracks.

The image in each of Figs. 6(e), 7(e), 8(e) and 9(e) exhibits a fully developed crack in its propagation phase. The corresponding bottom plate shows the histogram of the probability distribution that is significantly different from those in earlier cycles in plate pairs (a)–(d), which shows further gain in the information on crack damage. The image in each of Figs. 6(f) and 7(f) exhibits a completely broken specimen at ∼80 kilocycles for low-cycle fatigue. Similarly, the image in each of Figs. 8(f) and 9(f) exhibits a completely broken specimen at ∼240 kilocycles for high-cycle fatigue. The corresponding bottom plates show the delta distribution indicating complete information on crack damage.

The observation in Figs. 6–9 is further clarified by using the notion of entropy (see Eq. (2)). The data at the nominal condition have been partitioned using the maximum entropy principle, which leads to uniform probability distribution as seen in the bottom plate of plate pair (a) in Figs. 6–9. In contrast, for the completely broken stage of the specimen, the entire probability distribution is concentrated in only one state as seen in the bottom plate of plate pair (f) in Figs 6–9, due to very large attenuation of the ultrasonic signal. This phenomenon of the test specimen being completely broken signifies certainty of information and hence zero entropy. Therefore, as the fatigue crack damage evolves, the uniform distribution (i.e., maximum entropy) under nominal condition degenerates toward the delta distribution (i.e., zero entropy) for the broken specimen. In the intermediate stages, gradual degradation can be quantitatively evaluated using this information.

Fig. 10 shows the evolution of anomaly measure profiles obtained by STSA of ultrasonic data for low-cycle and high-cycle fatigue and the corresponding crack growth plots. The top plate on the left column in Fig. 10 shows profiles of anomaly measure for both left and right sides of the notch under low-cycle fatigue. Similarly, the top plate on the right column shows profiles of anomaly measure for both left and right sides of the notch under high-cycle fatigue. The bottom plates show the corresponding crack growth profiles. In each curve in the top plates of Fig. 10, the sharp change in the slope of the anomaly measure indicates the onset of crack propagation phase. This

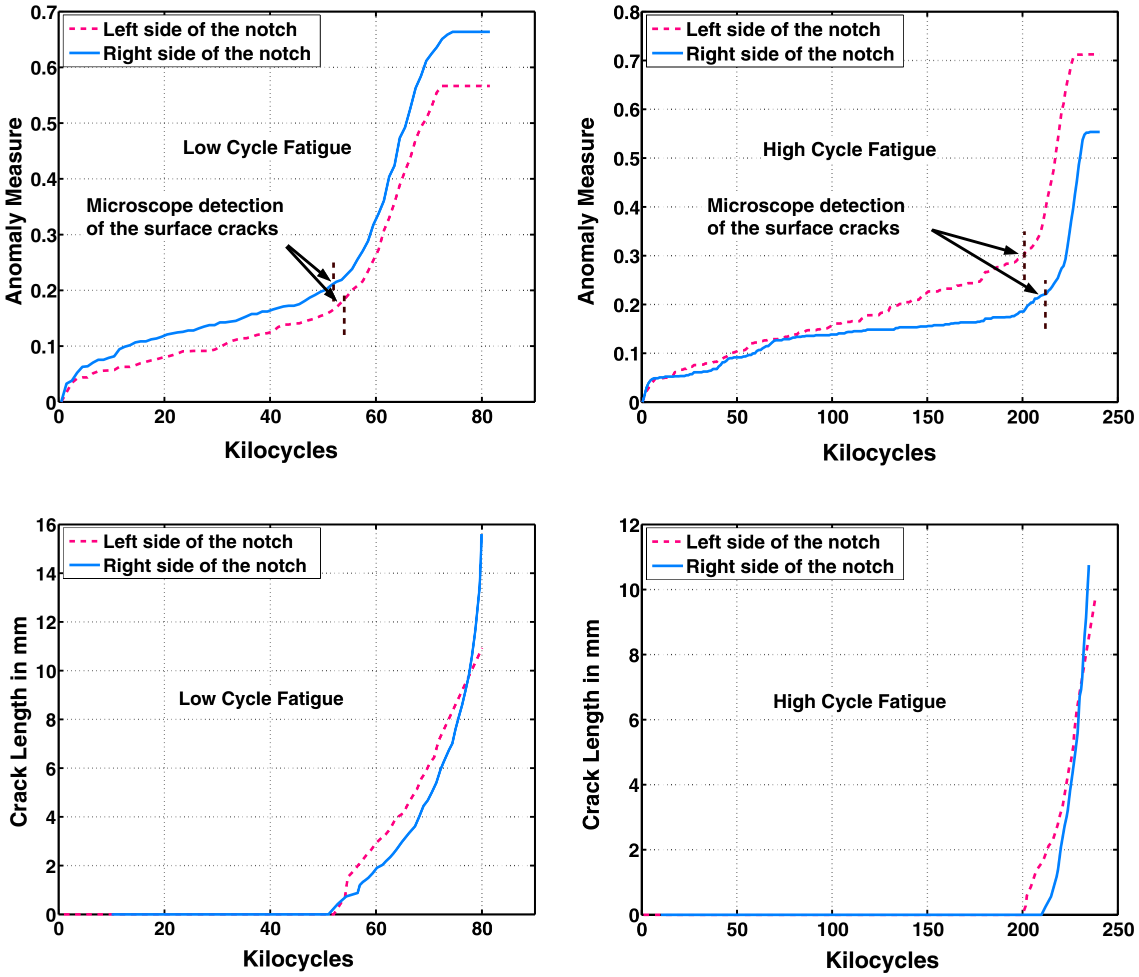


Fig. 10. Profiles of anomaly measure and crack growth for center-notched specimens under low cycle and high-cycle fatigue.

occurs approximately upon appearance of a surface crack. The vertical dashed lines in the top two plates of Fig. 10 indicate the first detection of a surface crack by the optical microscope.

It is observed that small changes can be detected by STSA significantly before the microscope can capture a surface crack. The slope of the anomaly measure represents the fatigue damage growth rate while the magnitude indicates the accumulated fatigue damage starting from the nominal condition. An abrupt increase in the slope (i.e., a sharp rise in the curvature) of anomaly measure profile provides a clear insight into a forthcoming failure. The region to the right of the vertical dashed lines for left side and right side anomaly measure curves in the top plate on left column of Fig. 10 can be considered to be the boundary of the crack initiation and crack propagation phases, where the growth of anomaly measure is significantly faster than that in the crack initiation phase. However, the critical information lies in the region towards the left of the vertical lines which is broadly identified as crack initiation phase where no crack was visible on the surface. This is the region where micro-structural damage such as multiple small cracks possibly caused small changes in the ultrasonic signal profile. After the multiple small cracks coalesce together to form a single large crack, the crack propagation phase starts. Damage evolution undergoes elastoplastic deformation under low-cycle fatigue due to a large number of dislocation movements and multiple small cracks develop from a very early stage. Similar effects are observed for high-cycle fatigue as seen in the top plate on right column of Fig. 10. In the region towards the right of the vertical dashed lines, anomaly measure profiles show significantly faster growth as compared to the slow growth during crack initiation towards the left of the vertical lines. For low-cycle fatigue, a relatively large slope in the anomaly measure is observed even during crack initiation region indicating occurrence of significant damage during this phase. Crack initiation prevailed for ~52–55 kilocycles for low-cycle fatigue, which is ~68% of the total life. For high-cycle fatigue, crack initiation prevailed for a relatively longer period of ~205–215 kilocycles for development of a single large crack, which amounts to ~87% of the total life.

The top two plates in Fig. 10 show a relatively large slope of anomaly measure from the start of cyclic loading to ~5 kilocycles for both low-cycle and high-cycle fatigue.

This is the stage where micro-structural damage (e.g., due to dislocation movements and accumulation, and persistent slip band formation) induces hardening of the strained components [45] and changes the ultrasonic impedance leading to a sharp rise in the local slope of the anomaly measure profile. After these initial effects subside, a modest reduction of the slope takes place for the remaining part of the crack initiation phase. A sharp rise of slope is again observed at the onset of the crack propagation phase. Similar phenomena were observed by Berkovits and Fang [47] in acoustic emission experiments on smooth specimens of Incoloy 901 at room temperature. Further experiments and micro-structural analysis are necessary to confirm these findings.

7. Summary, conclusions, and future work

This paper presents the concept, theory, and experimental validation of a statistical pattern identification tool for early detection and online monitoring of fatigue damage in polycrystalline alloys. The underlying principle of fatigue damage detection is built upon symbolic time series analysis of ultrasonic sensor signals. A combination of maximum-entropy partitioning in the wavelet domain and symbolic dynamics enables fatigue damage detection significantly before the onset of crack propagation. The codes of damage analysis are executable in real time and have been demonstrated on a special-purpose fatigue testing apparatus on 7075-T6 aluminum alloy specimens under: stress controlled low-cycle fatigue and stress controlled high-cycle fatigue. The results consistently indicate that the code is capable of detecting damage before any surface cracks are captured by the optical microscope.

The reported work is a step toward building a reliable instrumentation system for early detection of fatigue damage in polycrystalline alloys. Further experimental, analytical and micro-structural research is necessary before its usage in industry. While there are many technical issues that need to be addressed, the following research topics are being currently pursued.

- Statistical analysis of an ensemble of ultrasonic time series data sets, collected under identical loading and environmental conditions, to account for manufacturing and material uncertainties.
- Microstructural analysis to investigate the early stages of fatigue damage.
- Validation of the STSA technique for fatigue damage monitoring under different conditions, such as variable-amplitude block loading and spectral loading.

References

[1] Ozekici S. Reliability and maintenance of complex systems, vol. 154, NATO Advanced Science Institutes (ASI) Series F: Computer and Systems Sciences, Berlin, Germany, 1996.

[2] Meggiolaro M, Castro J. Statistical evaluation of strain-life fatigue crack initiation predictions. Int J Fatigue 2004;26:463–76.

[3] Ishihara S, McEvily A. Analysis of short fatigue crack growth in cast aluminium alloys. Int J Fatigue 2002;24:1169–74.

[4] Bjerkén C, Melin S. A tool to model short crack fatigue growth using a discrete dislocation formulation. Int J Fatigue 2003;25:559–66.

[5] Ramsamooj D. Analytical prediction of short to long fatigue crack growth rate using small- and large-scale yielding fracture mechanics. Int J Fatigue 2003;25:923–33.

[6] Pathria R. Statistical mechanics. Elsevier Science and Technology Books; 1996.

[7] Sobczyk K, Spencer B. Random fatigue: Data to theory. Boston, MA: Academic Press; 1992.

[8] Ott E. Chaos in dynamical systems. Cambridge University Press; 1993.

[9] Keller E, Ray A. Real time health monitoring of mechanical structures. Struct Health Monitor 2003;2(3):191–203.

[10] Grondel S, Delebarre C, Assaad J, Dupuis J, Reithler L. Fatigue crack monitoring of riveted aluminium strap joints by lamb wave analysis and acoustic emission measurement techniques. NDT & E Int 2002;35:137–46.

[11] Cook D, Berthelot Y. Detection of small surface-breaking fatigue cracks in steel using scattering of rayleigh waves. NDT & E Int 2001;34:483–92.

[12] Zilberstein V, Walrath K, Grundy D, Schlicker D, Goldfine N, Abramovici E, et al. Mwm eddy-current arrays for crack initiation and growth monitoring. Int J Fatigue 2003;25:1147–55.

[13] Witney A, Li YF, Wang J, Wang MZ, DeLuccia JJ, Laird C. Electrochemical fatigue sensor response to Ti-6 wt.% Al-4 wt.% v and 4130 steel. Philos Mag 2004;84(3–5):331–49.

[14] Yang B, Liaw PK, Wang G, Peter WH, Buchanan R, Yokoyama Y, et al. Thermal-imaging technologies for detecting damage during high-cycle fatigue. Metall Mater Trans A 2004;35A:15–24.

[15] Baram J. Fatigue-life prediction by an order statistics treatment of acoustic-emission signals. Exp Mech 1993;33:189–94.

[16] Harris DO, Dunegan HL. Continuous monitoring of fatigue–crack growth by acoustic-emission techniques. Exp Mech 1974;14:71–80.

[17] Lee KY. Cyclic ae count rate and crack growth rate under low cycle fatigue fracture loading. Eng Fracture Mech 1989;34(5/6):1069–73.

[18] Lysak MV. Development of the theory of acoustic emission by propagating cracks in terms of fracture mechanics. Eng Fracture Mech 1996;55(3):443–52.

[19] Scala M, Cousland SM. Acoustic emission during fatigue crack propagation in the aluminium alloys 2024 and 2124. Mater Sci Eng 1983;61:211–8.

[20] Zilberstein V, Grundy D, Weiss V, Goldfine N, Abramovici E, Newman J, et al. Early detection and monitoring of fatigue in high strength steels with mwm-arrays. Int J Fatigue 2005;27:1644–52.

[21] Bai HS, Yu LY, He (Ho) JW. A monitoring system for contact fatigue crack testing. NDT Int 1989;22(3):162–7.

[22] Yusa N, Janousek L, Rebican M, Chen Z, Miya K, Chigusa N, et al. Detection of embedded fatigue cracks in inconel weld overlay and the evaluation of the minimum thickness of the weld overlay using eddy current testing. Nucl Eng Des 2006;236(18):1852–9.

[23] Anson L, Chivers R, Puttick K. On the feasibility of detecting precracking fatigue damage in metal matrix composites by ultrasonic techniques. Compos Sci Technol 1995;55:63–73.

[24] Vanlanduit S, Guillaume P, Linden G. Online monitoring of fatigue cracks using ultrasonic surface waves. NDT & E Int 2003;36:601–7.

[25] Rokhlin S, Kim JY. In situ ultrasonic monitoring of surface fatigue crack initiation and growth from surface cavity. Int J Fatigue 2003;25:41–9.

[26] Kenderian S, Berndt T, Green R, Djordjevic B. Ultrasonic monitoring of dislocations during fatigue of pearlitic rail steel. Mater Sci Eng 2003;348:90–9.

[27] Daw C, Finney C, Tracy E. A review of symbolic analysis of experimental data. Rev Sci Instrum 2003;74(2):915–30.

[28] Ray A. Symbolic dynamic analysis of complex systems for anomaly detection. Signal Process 2004;84(7):1115–30.

[29] Chin S, Ray A, Rajagopalan V. Symbolic time series analysis for anomaly detection: A comparative evaluation. Signal Process 2005;85(9):1859–68.

[30] Gupta S, Ray A, Keller E. Symbolic time series analysis of ultrasonic data for early detection of fatigue damage. Mech Syst Signal Process. Available from: www.ScienceDirect.com, in press.

[31] Rajagopalan V, Ray A. Symbolic time series analysis via wavelet-based partitioning. Signal Process 2006;86(1):3309–20.

[32] Lind D, Marcus M. An introduction to symbolic dynamics and coding. United Kingdom: Cambridge University Press; 1995.

[33] Badii R, Politi A. Complexity hierarchical structures and scaling in physics. United Kingdom: Cambridge University Press; 1997.

[34] Hopcroft H, Motwani R, Ullman J. Introduction to automata theory, languages, and computation. 2nd ed. Boston: Addison Wesley; 2001.

[35] Abarbanel H. The analysis of observed chaotic data. New York: Springer-Verlag; 1996.

[36] Davidchack R, Lai Y, Bolt E, Dhamala H. Estimating generating partitions of chaotic systems by unstable periodic orbits. Phys Rev E 2000;61:1353–6.

[37] Kennel M, Buhl M. Estimating good discrete partitions form observed data: Symbolic false nearest neighbors. Phys Rev E 2003;91(8):084–102.

[38] Mallat S. A wavelet tour of signal processing 2/e. Academic Press; 1998.

[39] Cover TM, Thomas JA. Elements of information theory. New York: John Wiley; 1991.

[40] Naylor AW, Sell GR. Linear operator theory in engineering and science. New York: Springer-Verlag; 1982.

[41] Keller EE, Real time sensing of fatigue crack damage for information-based decision and control. PhD thesis, 2001, Department of Mechanical Engineering, Pennsylvania State University, State College, PA.

[42] Nagy P. Fatigue damage assessment by nonlinear ultrasonic materials characterization. Ultrasonics 1998;36:375–81.

[43] Cantrell J, Yost W. Nonlinear ultrasonic characterization of fatigue microstructures. Int J Fatigue 2001;23:487–90.

[44] Rose J. Ultrasonic waves in solid media. Cambridge University Press; 2004.

[45] Suresh S. Fatigue of materials. Cambridge, UK: Cambridge University Press; 1998.

[46] Toolbox Wavelets, MATLAB. Mathworks Inc.

[47] Berkovits A, Fang D. Study of fatigue crack characteristics by acoustic emission. Eng Fracture Mech 1995;51(3): 401–16.

Available online at www.sciencedirect.com
SCIENCE DIRECT®

Physica A 359 (2006) 1–23

PHYSICA A

www.elsevier.com/locate/physa

Correlation regimes in fluctuations of fatigue crack growth

Nicola Scafetta[a,*], Asok Ray[b], Bruce J. West[a,c]

[a]*Department of Physics, Duke University, Durham, NC 27708, USA*
[b]*Mechanical Engineering Department, The Pennsylvania State University, University Park, PA 16802, USA*
[c]*Mathematics Division, Army Research Office, Research Triangle Park, NC 27709, USA*

Received 22 February 2005
Available online 1 June 2005

Abstract

This paper investigates correlation properties of fluctuations in fatigue crack growth of polycrystalline materials, such as ductile alloys, that are commonly encountered in structures and machinery components of complex electromechanical systems. The model of crack damage measure indicates that the fluctuations of fatigue crack growth are characterized by strong correlation patterns within short-time scales and are uncorrelated for larger time scales. The two correlation regimes suggest that the 7075-T6 aluminum alloy, analyzed in this paper, is characterized by a micro-structure which is responsible for an intermittent correlated dynamics of fatigue crack growth within a certain scale. The constitutive equations of the damage measure are built upon the physics of fracture mechanics and are substantiated by Karhunen–Loève decomposition of fatigue test data. Statistical orthogonality of the estimated damage measure and the resulting estimation error are demonstrated in a Hilbert space setting.

1. Introduction

The fracture of solids and the growth of cracks is a typical instability phenomena which are known to be strongly nonlinear. Herein we apply to fracture mechanics

[*]Corresponding author. Tel.: +1 919 660 2643; fax: +1 919 660 2671.
E-mail address: nicola.scafetta@duke.edu (N. Scafetta).

0378-4371/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.physa.2005.03.052

some of the recent methods developed in statistical physics. In particular, we use the notion of fractal statistics to describe the correlation of the fluctuations around fatigue crack growth in polycrystalline materials, such as ductile alloys. In this paper, we have investigated the fatigue fracture properties of 7075-T6 aluminum alloy.

The importance of this investigation is that, in both physics and engineering literature, the fluctuations around fatigue crack growth in a typical material have always been assumed to be random or uncorrelated noise. Consequently, the associated models include uncorrelated random processes. For example, in agreement with the existent theory of micro-level fatigue cracking, Bogdanoff and Kozin [1] proposed a Poisson-like uncorrelated-increment jump model of fatigue crack phenomena. An alternative approach to stochastic modeling of fatigue crack damage is to randomize the coefficients of an existing deterministic model to represent material inhomogeneity [2]. A third approach has been to adopt a deterministic model of fatigue crack growth in addition to a random process, see for example Refs. [3–5].

The fatigue crack growth process can also be modeled by nonlinear stochastic differential equations using Itô statistics [6] that again presuppose randomness of the fluctuations. Specifically, the Kolmogorov forward and backward diffusion equations, which require solutions of nonlinear partial differential equations, have been proposed to generate the statistical information required for risk analysis of mechanical structures [7,8]. These nonlinear partial differential equations have only been solved numerically and the numerical procedures are computationally intensive as they rely on fine-mesh models using finite-element or combined finite-difference and finite-element methods [9]. Casciati et al. [10] have analytically approximated the solution of the Itô equations by Hermite moments to generate a probability distribution function of the crack length.

Several studies have determined that the stochastic fluctuations observed in innumerable natural phenomena are not simply random, i.e., uncorrelated noise, but present correlation patterns that reveal complex and alternative dynamics and/or material microstructures. Thus, the purpose of the present research is to determine whether uncorrelated stochastic models such as those previously discussed in the literature are realistic in describing the fluctuations around fatigue crack growth in polycrystalline materials, or whether such fluctuations present patterns that would reveal complex material micro-structure requiring alternative correlated stochastic models. Two main classes of correlation patterns are commonly observed in natural time series and these are denoted as short- and long-time correlations. Short-time correlations are characterized by phenomena that rapidly lose memory of past or distant events. This happens, for example, when the autocorrelation function of the time series decays exponentially in the time separation between two elements. By contrast, long-time correlations are characterized by autocorrelation functions that decay more slowly than (negative) exponentials; one example is the inverse power-law decay.

A simple model, which has been extensively used in the interpretation of stochastic fluctuations in a time series $\{\xi_i\}$ with $i = 1, 2, \ldots, N$, is based on the evaluation of the mean-square displacement of the diffusion-like processes generated by trajectories

$X_n(t)$ defined as

$$X_n(t) = \sum_{j=1}^{t} \xi_{n+j} \, . \tag{1}$$

If $\{\xi_i\}$ is a white random sequence, the diffusion process is a well-known Brownian motion. The central limit theorem applied to the diffusion distribution generated by trajectories $X_n(t)$ yields a probability density that converges to a Gaussian function whose mean-square displacement converges asymptotically to

$$\langle X(t)^2 \rangle \propto t^{\alpha} \, , \tag{2}$$

with $\alpha = 1$. In general, it is possible to have anomalous behavior yielding enhanced diffusion ($\alpha > 1$) that has been known for 20 years to arise in dynamically chaotic systems [11], or sublinear diffusive growth ($\alpha < 1$) that is familiar from disordered fractal materials [12].

Anomalous diffusion reveals persistent (for an enhanced diffusive growth) or antipersistent (for a sublinear diffusive growth) correlation patterns in the dynamics of a random walk. A persistent random walk is characterized by a probability of stepping in the direction of the previous step that is greater than that of reversing directions. An antipersistent random walk is characterized by a probability of stepping in the direction of the previous step that is less than that of reversing directions. Sometimes a momentarily initial enhanced or sublinear diffusive growth, lasting up to a certain time-scale, is generated by the statistical transition to the asymptotic regime of the diffusion process. For example, a simple discrete random walk is described by a binomial distribution that only asymptotically converges to a Gaussian while initially presenting an enhanced diffusive growth [13]. Thus, a real autocorrelated time series will lose its correlation patterns if the temporal order of the sequence is randomized.

There are a number of different theoretical approaches that explain the anomalous diffusion depicted in (2). One such explanatory model is that of an infinitely long correlated random walk in which $\alpha = 2H$, where H is the Hurst exponent in the interval $0 \leqslant H \leqslant 1$ with the case $H = 0.5$ corresponding to a simple random walk. This model has been used extensively in the interpretation of fluctuations in time series in the physical and life sciences [14] and is called fractional Gaussian noise [15]. Another kind of anomalous diffusion has to do with taking steps that are uncorrelated in time, but on a random or fractal, not a regular lattice. In the second model, an anomalous diffusion occurs because geometrical obstacles exist on all length scales and such obstacles inhibit transport. Havlin and Ben-Avraham [12] point out that the anomalous exponent α is related to the fractal dimension of the random walk path on the lattice. There is a third possible explanation of the anomaly in (2) called a Lévy walk [16] that was first used to understand turbulent diffusion [16] and yields $\alpha \approx 3$, which is consistent with Richardson's law of enhanced diffusion [17].

Physical examples of anomalous diffusion processes are earthquakes [18], rainfall [16,19], turbulent fluid flow [20], relaxation of stress in viscoelastic materials [14,21], solar flares [22–24], and other processes with slip-stick dynamics. Recently, a multiscaling comparative analysis to distinguish Lévy walk intermittent noise from fractal Gaussian intermittent noise was suggested by Scafetta and West [25].

Finally, a physical system might be characterized by different values of the scaling exponent α at different scales [26]. Usually, this means that one system is characterized by a non-self-affine structure. The scale at which the transition from one scaling regime to another occurs indicates the scale at which the structure changes. In this work we determine that the fluctuations around the ballistic growth of fatigue cracks in ductile alloys present such a scale transition from a strongly correlated regime at short-time scales to a random regime at longer time-scales. Properties, such as grain size distribution, degree of heterogeneity, the existence of microscopic defects, inclusions, twin boundaries and dislocations, of polycrystalline materials may contribute to the micro-mechanisms of fatigue fracture revealed by the present analysis.

This paper is organized into six sections, including the present one, and an appendix. Section 2 provides the underlying phenomenology of the stochastic damage measure. Section 3 presents Karhunen–Loève (KL) decomposition of fatigue test data to formulate an estimate of the stochastic measure, which is statistically orthogonal to the estimation error. Section 4 focuses on identification of the model parameters and their probability distributions. Section 5 presents the results of model prediction by Monte Carlo simulation. The paper is summarized and concluded in Section 6 with recommendations for future research.

2. Measure of fatigue crack damage

Traditionally fatigue crack growth models have been formulated by fitting estimated mean values of fatigue crack length $\hat{a}_t$, generated from ensemble averages of experimental data, as functions of time in units of cycles [27,28]. Ray and Patankar [29] have formulated the state-space modeling concept of crack growth based on fracture-mechanistic principles of the crack-closure concept [30]. The state-space model has been validated by fatigue test data for variable-amplitude cyclic loading, see for example Refs. [28,31,32].

The three panels in Fig. 1 show test data of cumulative fatigue crack growth in the 7075-T6 aluminum alloy under different cyclic loading [33]. It is important to note that the crack growth curves do not increase smoothly, but they exhibit fluctuations around an ideal smooth curve of crack growth representing ballistic growth. In this context, a major objective of the paper is to investigate the autocorrelation properties of these fluctuations with the smooth curve removed. In the following we briefly review the theory and the standard phenomenological equations that describe the fatigue crack growth.

In linear fracture mechanics, it is assumed that the stressed material remains elastic and undamaged everywhere, except in a small domain in the vicinity of the crack tip. However, this view is not confirmed by experimental evidence and the process of fatigue damage accumulation could occur throughout the stressed volume. Paris and Erdogan

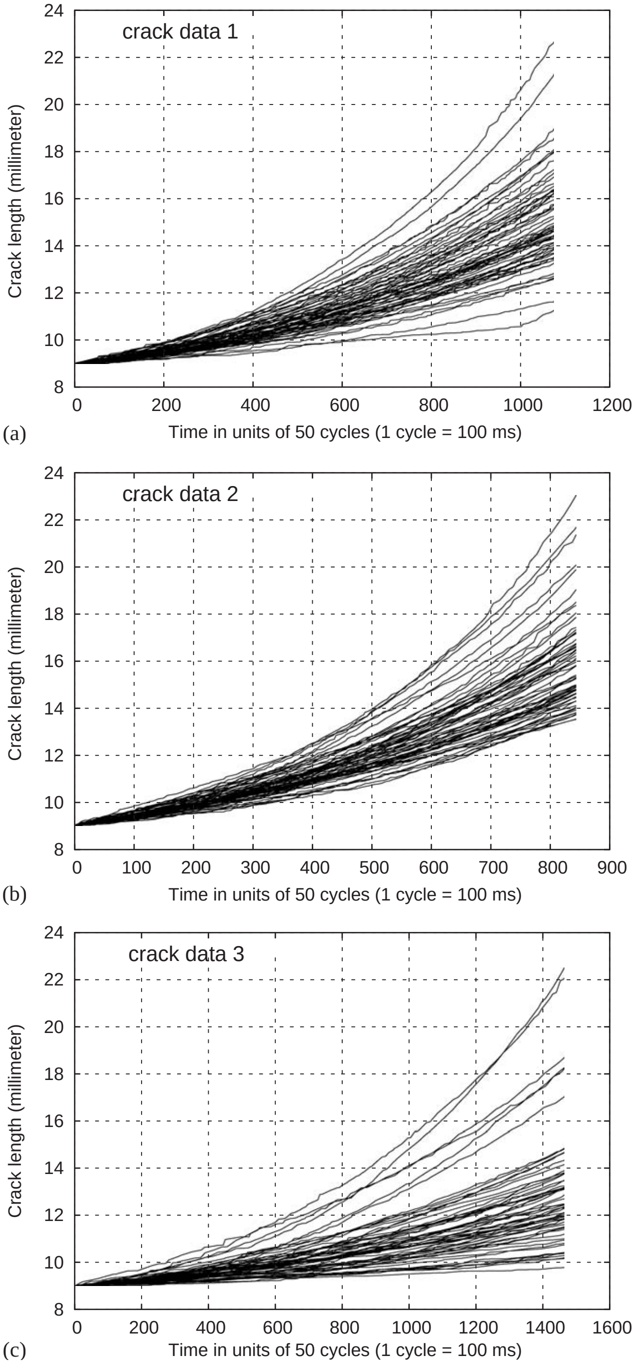


Fig. 1. Experimental data of 7075-T6 aluminum alloy. (a) $R = 0.6$ and Max stress = 70.65 MPa; (b) $R = 0.6$ and Max stress = 69.00 MPa; (c) $R = 0.6$ and Max stress = 47.09 MPa.

[27] originally developed a phenomenological model of crack growth rate, which depends on the stress history and is thus represented by a continuum rate equation having the hereditary structure. This model has been subsequently modified by many researchers (see, for example, citations in Refs. [34–36]) in the following form:

$$\delta\hat{a}_t \equiv \hat{a}_t - \hat{a}_{t-\delta t} = h(\Delta K_t^{eff})\delta t\,, \tag{3}$$

with $h(0) = 0$ and $\hat{a}_{t_0} > 0$ for $t \geqslant t_0$, where $\hat{a}_t$ is the estimated mean of the crack length at time t during a stress cycle and δt is the time duration of the stress cycle; and ΔK_t^{eff} is the stress intensity factor range at time t, which is given by the experimentally validated empirical model

$$\Delta K_t^{eff} = \Delta S_t \sqrt{\pi \hat{a}_{t-\delta t}} F(\hat{a}_{t-\delta t})\,, \tag{4}$$

where ΔS_t is the range (i.e., the difference between maximum and minimum values) of the stress cycle at time t, which is directly related to the applied load. Experimental observations suggest that both duration and shape of a stress cycle are not relevant for crack growth in ductile alloys at room temperature. A stress cycle is only characterized by the minimum stress $S^{\min}$ and the maximum stress $S^{\max}$, respectively, and is denoted as the ordered pair $(S^{\min}, S^{\max})$. The empirical relation $F(\bullet)$ in Eq. (4) represents the geometry of the crack tip; for center-cracked specimens of half-width w with $0 < \hat{a}_t < w$ at all $t \geqslant t_0$, the structure of $F(\bullet)$ has been experimentally determined as [35]

$$F(\hat{a}_{t-\delta t}) = \sqrt{\sec\left(\frac{\pi}{2w}\,\hat{a}_{t-\delta t}\right)}\,. \tag{5}$$

The function $h(\bullet)$ in Eq. (3) is a non-negative Lebesgue-measurable function that is dependent on the material and geometry of the stressed component. It has been shown in the fracture mechanics literature [35,36] that, for center-cracked specimens of ductile alloys, the function $h(\bullet)$ obeys the power law:

$$h(\Delta K_t^{eff}) = (\Delta K_t^{eff})^m\,, \tag{6}$$

where the exponent parameter m is dependent on the material of the stressed component; for ductile alloys, m is in the range of 2.5–5.0 [35].

Eqs. (3)–(6) are now combined to formulate a mean-value model of fatigue crack growth for center-cracked specimens of ductile alloy materials:

$$\delta\hat{a}_t \propto \left[\Delta S_t \sqrt{\hat{a}_{t-\delta t}\sec\left(\frac{\pi}{2w}\,\hat{a}_{t-\delta t}\right)}\right]^m \delta t\,, \tag{7}$$

with $\hat{a}_{t_0} > 0$ and $t \geqslant t_0$.

Following Sobczyk and Spencer [9] and the pertinent references cited therein, we randomize the deterministic mean-value model, Eq. (7), to obtain a stochastic model for the rate of crack growth. The stochastic model of continuous crack length is built upon the model structure proposed by Ray [29,37], and is given by

$$\mathrm{d}c_t(\zeta) = \Omega(\zeta, t)\left[\Delta S_t \sqrt{\frac{c_t(\zeta)}{\cos((\pi/2)c_t(\zeta))}}\right]^m \mathrm{d}t \cong \Omega(\zeta, t)\frac{(\Delta S_t\sqrt{c_t(\zeta)})^m}{1 - m((\pi/4)c_t(\zeta))^2}\,\mathrm{d}t\,, \tag{8}$$

where the random sample ζ signifies a specimen or a machine component on which a fatigue test is conducted; the dimensionless stochastic crack length $c_t(\zeta)$ is normalized with respect to the half width w, i.e., the mean value $\hat{c}_t \equiv \hat{a}_t/w$. Eq. (8) is a continuous stochastic version of Eq. (7), where the differential of the stochastic crack length $\mathrm{d}c_t(\zeta)$ is a function of the crack length $c_t(\zeta)$ at time t and the normalized stress $\Delta S_t \equiv \Delta S_t^e/S^y$, where S^y is the yield stress of the material. The condition $0 < c_{t_0} \leqslant c_t < 4/\pi\sqrt{m}$ is imposed to ensure non-negativity of the crack length increment almost surely, i.e., $\mathrm{d}c_t(\zeta) > 0$ for almost all samples ζ. The stochastic process of crack growth is largely dependent on the second-order random process $\Omega(\zeta, t)$ and the exponent parameter m in Eq. (8).

To investigate the stochastic properties of the fatigue crack growth process, we separate $\Omega(\zeta, t)$ into two parts as

$$\Omega(\zeta, t) = \Omega_0(\zeta)[1 + \Omega_1(\zeta, t)]\,, \tag{9}$$

where the time-independent component $\Omega_0(\zeta)$ represents uncertainties in manufacturing, for example in machining, and makes a major contribution to the ballistic component of the crack growth; the time-dependent component $\Omega_1(\zeta, t)$ represents uncertainties in the material micro-structure and crack length measurements that may vary with crack propagation in a sample ζ. This latter component is primarily responsible for the small fluctuations around the ballistic component of crack growth whose autocorrelation properties we study.

We postulate that Ω_0 and Ω_1 in Eq. (9) are statistically independent of one another for all $t \geqslant t_0$, where t_0 is the initial time. The rationale for this independence assumption is that inhomogeneity of the material micro-structure and measurement noise, associated with each test specimen and represented by $\Omega_1(\zeta, t)$, are unaffected by the uncertainty $\Omega_0(\zeta)$ due, for example, to machining operations. Without loss of generality, we assume that the fluctuations in time have a zero mean value, i.e., $\langle \Omega_1(\zeta, t)\rangle = 0$ for all $t \geqslant t_0$. Furthermore, non-negativity of the crack growth rate $dc_t(\zeta)$ in Eq. (8) is assured in the almost sure (a.s.) sense by imposing the constraint $\Omega_0(\zeta) \geqslant 0$ with probability 1 (w.p. 1).

For notational brevity, let us suppress the term ζ in random processes like $c_t(\zeta)$ and $\Omega(\zeta, t)$. A combination of Eqs. (8) and (9) and few simple algebraic steps yield the following equation for each sample point ζ:

$$\left[c_t^{-m/2} - m\left(\frac{\pi}{4}\right)^2 c_t^{2-m/2}\right]\mathrm{d}c_t = (\Delta S_t)^m \Omega_0[1 + \Omega_1(t)]\,\mathrm{d}t \quad \text{w.p. } 1\,. \tag{10}$$

Pointwise integration of Eq. (10) yields the solution of fatigue damage increment from the initial time t_0 to the current time t as

$$\psi(t, t_0) = \int_{t_0}^{t} (\Delta S_{t'})^m \Omega_0[1 + \Omega_1(t')]\,\mathrm{d}t' \quad \text{w.p. } 1\,. \tag{11}$$

An explicit expression of the stochastic diffusion process $\psi(t, t_0)$ is obtained by integrating the left-hand side of Eq. (10) and is given by

$$\psi(t, t_0) \equiv \left[\frac{c_t^{1-m/2} - c_{t_0}^{1-m/2}}{1 - m/2}\right] - m\left(\frac{\pi}{4}\right)^2 \left[\frac{c_t^{3-m/2} - c_{t_0}^{3-m/2}}{3 - m/2}\right], \tag{12}$$

where $\psi(t, t_0)$ represents a dimensionless non-negative measure of fatigue crack damage increment from the initial instant t_0 to the current instant t as a function of the normalized crack length. The constant parameter m in (12) is in the range of 2.5–5 for ductile alloys and metallic materials ensuring that $(1 - m/2) < 0$ and $(3 - m/2) > 0$. The diffusion process $\psi(t, t_0)$ is almost surely continuous because it is a continuous function of the crack length process c_t w.p. 1. Both c_t and $\psi(t, t_0)$ are measurable functions although their (probability) measure spaces are different. In essence, the probability of $\psi(t, t_0)$, conditioned on the initial crack length c_{t_0}, leads to a stochastic measure of fatigue crack damage increment at the instant t starting from the initial instant t_0.

For a constant stress range ΔS, we carry out the time integration in Eq. (11) to obtain

$$\psi(t, t_0) = (\Delta S)^m [\Omega_0(t - t_0) + \Theta(t, t_0)], \tag{13}$$

where the second term on the right-hand side is the time integral

$$\Theta(t, t_0) \equiv \Omega_0 \int_{t_0}^{t} \Omega_1(t')\, \mathrm{d}t' . \tag{14}$$

Thus, the stochastic diffusion process $\psi(t, t_0)$ according to model (13) is given as the sum of a random component, linear in time, plus a time-fluctuating component proportional to the diffusion process $\Theta(t, t_0)$.

The objective is to validate the model in Eq. (11) by decomposing the damage increment measure $\psi(t, t_0)$ into two parts that are mutually statistically independent and, at the same time, equivalent to the two components of the right-hand side of Eq. (13). That is, we would like to obtain an estimate $\hat{\psi}(t, t_0)$ of the stochastic damage increment measure $\psi(t, t_0)$ and of the fluctuations $\tilde{\psi}(t, t_0)$ around $\hat{\psi}(t, t_0)$ from the initial instant t_0 to the current instant t such that

$$\psi(t, t_0) \overset{ms}{=} \hat{\psi}(t, t_0) + \tilde{\psi}(t, t_0) , \tag{15}$$

where $\hat{\psi}(t, t_0)$ is statistically equivalent to $\Delta S^m \Omega_0(t - t_0)$, and $\tilde{\psi}(t, t_0)$ is statistically equivalent to $(\Delta S)^m \Theta(t, t_0)$ of Eq. (13).

To test the validity of the above postulate that the two components of the multiplicative random process $\Omega_0(\zeta)$ and $\Omega_1(\zeta, t)$ in Eq. (9) are statistically independent, we require that the zero-mean estimation error $\tilde{\psi}(t, t_0)$ be statistically orthogonal to the estimate of the increment measure $\hat{\psi}(t, t_0)$ in the Hilbert space $L_2(P)$ defined by the probability measure P. As such $\hat{\psi}(t, t_0)$ is the best linear estimate of the stochastic diffusion process. Based on mean-square continuity of the damage measure $\psi(t, t_0)$, the next section elaborates on the model structure laid out in

Eq. (15). To this end, we analyze experimental data sets of random fatigue via KL decomposition [38–40] that guarantees the above statistical orthogonality among the components of the decomposition. In Section 4 we also use these experimental data sets to identify the model parameters.

3. Karhunen–Loève decomposition of experimental data

In this section we analyze fatigue test data via KL decomposition [40] to justify the model structure postulated in Eqs. (11) and (12). We use the experimental data of random fatigue crack growth in the 7075-T6 aluminum alloy [33] and conduct the tests under different constant load amplitudes at ambient temperature. For all experiments the half-width is $w = 50.8\,\text{mm}$, the initial crack length is $a_{t_0} = 9\,\text{mm}$, and, therefore, the initial dimensionless crack length is $c_{t_0} = a_{t_0}/w = 0.18$ with probability 1. The Ghonem data sets were generated for 60 center-cracked specimens each at three different constant load amplitudes: (i) Set #1 with peak nominal stress of 70.65 MPa (10.25 ksi) and stress ratio $R \equiv S^{\min}/S^{\max} = 0.6$ for 54,000 cycles, the effective stress range $\Delta S^e = 15.84\,\text{MPa}$; (ii) Set #2 with peak nominal stress of 69.00 MPa (10.00 ksi) and $R = 0.5$ for 42,350 cycles, and $\Delta S^e = 17.80\,\text{MPa}$; and (iii) Set #3 with peak nominal stress of 47.09 MPa (6.83 ksi), $R = 0.4$ for 73,500 cycles, and $\Delta S^e = 13.24\,\text{MPa}$. The three experimental data sets [33] are shown in the three panels of Fig. 1.

The KL decomposition requires the mean and covariance of the stochastic measure of damage increment $\psi(t, t_0)$ which are expressed as

$$\mu_\psi(t, t_0) \equiv \langle \psi(t, t_0) \rangle \,,$$

$$C_{\psi\psi}(t_1, t_2; t_0) \equiv \langle [\psi(t_1, t_0) - \mu_\psi(t_1, t_0)][\psi(t_2, t_0) - \mu_\psi(t_2, t_0)] \rangle \,. \tag{16}$$

The covariance function $C_{\psi\psi}(t_1, t_2; t_0)$ in Eq. (16) is continuous at $t_1 = t_2 = t$ for all $t \geqslant t_0$. Hence, the process $\psi(t, t_0)$ is mean-square (*ms*) continuous based on a standard theorem of mean-square calculus [38,39]. The mean and covariance are calculate for the 60 available center-cracked specimens in each case.

Since only finitely many data points at n discrete instants are available from experiments, an obvious approach to the analysis of the damage estimate is to discretize over the finite time horizons $[t_0, t]$ so that the stochastic process $\psi(t, t_0)$ becomes the n-dimensional random vector ψ. Consequently, the covariance function $C_{\psi\psi}(t_1, t_2; t_0)$ in Eq. (16) is reduced to a real semipositive-definite $(n \times n)$ symmetric matrix $\mathbf{C}_{\psi\psi}$. Since the experimental data were collected at sufficiently close intervals, $\mathbf{C}_{\psi\psi}$ contains pertinent information of the crack damage process. The n (real non-negative) eigenvalues of $\mathbf{C}_{\psi\psi}$ are ordered as $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$, with the corresponding eigenvectors, $\varphi^1, \varphi^2, \ldots, \varphi^n$, which form an orthogonal basis of $\Re^n$ for signal decomposition. The KL decomposition also ensures that the n random coefficients of the basis vectors are statistically orthogonal, i.e., they have zero mean and are mutually uncorrelated. These random coefficients form a random vector $\mathbf{X} \equiv [x_1, x_2, \ldots, x_n]^{\mathrm{T}}$ having the covariance matrix $\mathbf{C}_{XX} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$, leading to a decomposition of the discretized signal as

$$\psi \overset{ms}{=} \langle \psi \rangle + \sum_{j=1}^{n} x_j \phi^j \, . \tag{17}$$

It was observed by Ray [37] that the statistics of crack length are dominated by the random coefficient corresponding to the principal eigenvector (i.e., the eigenvector associated with the largest eigenvalue) and that the combined effects of the remaining eigenvectors are small. Therefore, the signal ψ in Eq. (17) is expressed as the sum of a principal part and a (zero-mean) residual part that are mutually statistically orthogonal:

$$\psi \overset{ms}{=} \underbrace{\langle \psi \rangle + x_1 \phi^1}_{\text{principal part}} + \underbrace{\sum_{j=2}^{l} x_j \phi^j}_{\text{residual part}} \, . \tag{18}$$

Thus, as Eq. (15) requires, the vector ψ is expressed as the sum of the principal and residual parts with equality in the mean square (*ms*) as

$$\psi \overset{ms}{=} \hat{\psi} + \tilde{\psi} \, , \tag{19}$$

where the principal part is the damage estimate

$$\hat{\psi} \equiv \langle \psi \rangle + x_1 \phi^1 \, , \tag{20}$$

the residual part is the estimation error representing the fluctuations around the mean damage estimate (20)

$$\tilde{\psi} \equiv \sum_{j=2}^{n} x_j \phi^j \, , \tag{21}$$

and the resulting (normalized) mean square error [40] is

$$\varepsilon_{rms}^2 \equiv \frac{Trace\{Cov[\psi - \hat{\psi}]\}}{Trace\{Cov[\psi]\}} = \frac{\sum_{j=2}^{n} \lambda_j}{\sum_{j=1}^{n} \lambda_j} \, . \tag{22}$$

The KL decomposition of fatigue test data sets reveals that $0.01 \leqslant \varepsilon_{rms}^2 \leqslant 0.1$ for all three data sets.

The principal eigenvector $\phi^1(t)$, associated with the largest eigenvalue λ_1, closely fits the ramp function $(t - t_0)$ for each of the three data sets in Fig. 1; this is shown in Fig. 2 for the data set 1. Comparing the terms on the right-hand side of the discrete model in Eq. (19) with those of the continuous model in Eq. (13), it is reasonable to have the random variable $\Delta S^m[\Omega_0 - \mu_0]$ equal (in *ms* sense) to the random coefficient x_1 of the principal eigenvector $\varphi^1(t)$. Applying the lemma from the appendix, a mean-square equivalence between the KL decomposition model in Eq. (19) derived from the test data and the postulated model in Eq. (17) is established as

$$\underbrace{\langle \psi(t) \rangle}_{\text{discrete model (test data)}} \overset{ms}{\approx} \underbrace{\Delta S^m \mu_0 (t - t_0)}_{\text{continuous model (constitutive relation)}} \, , \tag{23}$$

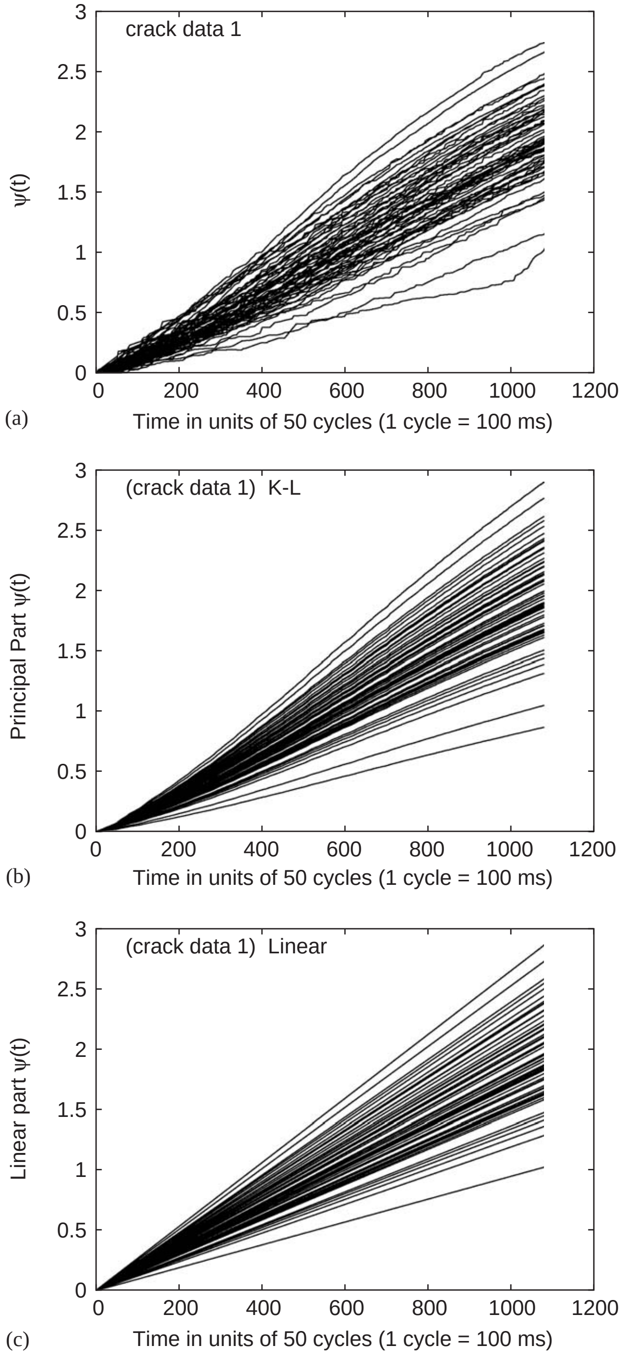


Fig. 2. (a) Curves $\psi(t)$ obtained with Eq. (12) for the experimental data of 7075-T6 aluminum alloy for set #1. The value of m used is $m = 4.0$. (b) Principal part of the KL decomposition against (c) the linear approximation of the continuous model made in Eqs. (23) plus (24) of the curves $\psi(t)$.

$$\underset{\text{discrete model (test data)}}{x_1\phi^1(t)} \quad \overset{ms}{\approx} \quad \underset{\text{continuous model (constitutive relation)}}{\Delta S^m[\Omega_0 - \mu_0](t - t_0)} \quad , \tag{24}$$

$$\underset{\text{discrete model (test data)}}{\sum_{j=2}^{n} x_j\phi^j} \quad \overset{ms}{\approx} \quad \underset{\text{continuous model (constitutive relation)}}{(\Delta S)^m\Theta(t, t_0)} \quad . \tag{25}$$

Thus, we have $\hat{\psi} = \langle\psi(t)\rangle + x_1\phi^1(t) \approx \Delta S^m\Omega_0(t - t_0)$, and $\tilde{\psi} = \sum_{j=2}^{n} x_j\phi^j \approx \Delta S^m\Theta(t, t_0)$ as assumed in Eq. (15). The two entities on the left-hand side in Eqs. (24) and (25) are mutually statistically orthogonal by construction. Similarly, in view of Eq. (15), the zero-mean estimation error $\tilde{\psi}(t, t_0)$ is statistically orthogonal to $\hat{\psi}(t, t_0)$ in the Hilbert space $L_2(P)$ defined by the probability measure P associated with the stochastic process $\psi(t, t_0)$. As such $\hat{\psi}(t, t_0)$ can be viewed as the best linear estimate of $\psi(t, t_0)$ with the least error $\tilde{\psi}(t, t_0)$ in the mean-square sense.

It follows from Eqs. (15) to (25) that the uncertainties associated with an individual sample resulting from the damage measure estimate $\hat{\psi}(t, t_0)$ dominate the cumulative effects of material inhomogeneity and measurement noise in the estimation error $\tilde{\psi}(t, t_0)$ unless $(t - t_0)$ is small. Therefore, from the perspectives of material-health monitoring, risk analysis, and remaining life prediction where the inter-maintenance interval $(t - t_0)$ is expected to be large, a reasonably accurate identification of the mean μ_0 and variance σ_0^2 of the random parameter Ω_0 is crucial, while the role of the diffusion process $\Theta(t, t_0)$ is relatively less significant. This observation is consistent with the statistical analysis of fatigue test data by Ditlevsen [2], where the random process described by Eq. (25) is treated as the zero-mean residual. Ditlevsen [2] also observed largely similar properties by statistical analysis. Nevertheless, the stochastic properties of fluctuating function $\Theta(t, t_0)$, which we investigate, can disclose important information about the material structure of alloys during crack damage.

4. Data analysis

In this section we investigate the stochastic equivalence made in Eq. (25) between the residual component of the signal as obtained by the KL decomposition and the linear approximation. The first step is to evaluate the exponent parameter m by fitting the data of the crack growth with Eq. (8). The fit is done by considering the crack increments from all 60 cases for each of the three experiments.

By using the empirical values of m it is possible to estimate $\psi(t, t_0)$ via Eq. (12). The three plots in Fig. 2 compare the curve $\psi(t, t_0)$, its principal part according to the KL decomposition and its linear approximation according to the continuous model made in Eqs. (23) plus (24) for set #1: the figures for the other data sets look qualitatively similar. Fig. 3 shows the quality of the equivalence made in Eqs. (23) plus (24) between the discrete model, which makes use of the KL decomposition, and the continuous model, which makes use of a linear approximation.

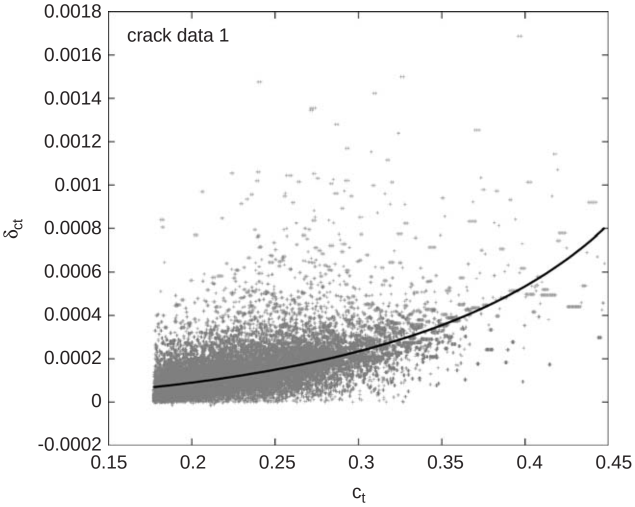


Fig. 3. Increments δc_t against crack length c_t fit with Eq. (8) (solid curve) for the experimental data of 7075-T6 aluminum alloy for the set #1: $\Omega\Delta S^m = 0.0019 \pm 0.0002$ and $m = 4.0 \pm 0.2$.

Fig. 3 shows the fitted data and the results for set #1; the figures for the other sets are similar. The parameters for all the three sets are listed below.

- $\Omega(\zeta)\Delta S^m = 0.0019 \pm 0.0002$ and $m = 4.0 \pm 0.2$ for set #1;
- $\Omega(\zeta)\Delta S^m = 0.0022 \pm 0.0002$ and $m = 3.8 \pm 0.2$ for set #2;
- $\Omega(\zeta)\Delta S^m = 0.0018 \pm 0.0002$ and $m = 4.7 \pm 0.2$ for set #3.

4.1. Diffusion standard deviation analysis of the fluctuations

We evaluate the stochastic equivalence made in Eq. (25) between the residual part of the discrete model, which makes use of the KL decomposition, and the residual part of the continuous model, which makes use of a linear approximation, in two steps. Step 1 compares the size of the increments of the correspondent residual parts; and Step 2 adopts the standard deviation analysis (SDA) which is a statistical formalism to study the long-time correlation in a fractal time series.

Because $\Theta(t) = \textit{residual part}$, the increments are given by $\theta_t = \Theta(t) - \Theta(t-1)$. We calculate the standard deviation, σ_θ, of the increments $\{\theta_t\}$ for each residual component estimated by means of the KL decomposition and of the linear approximation, respectively. Finally, we calculate the average of the standard deviation, $\langle\sigma_\theta\rangle$, between the 60 σ_θ for each of the three cases. The results shown in Table 1 demonstrate the compatibility of the increments obtained with the residual parts of the KL decomposition and the residual part of the continuous model.

Now, let us suppose that a generic residual curve is given by the function $\Theta(t)$, see Eq. (25), that in this specific case is a kind of random walk around the ballistic part

Table 1
Values of the fitting parameters μ and σ of the lognormal distribution (29) of the histograms shown in Fig. 7

	μ	σ
Set #1	0.58 ± 0.05	0.20 ± 0.02
Set #2	0.74 ± 0.05	0.16 ± 0.02
Set #3	0.42 ± 0.05	0.45 ± 0.04

of the signal, which is the principal component of the KL decomposition or the linear component of the continuous model. The SDA determines the scaling of the standard deviation of the diffusion process defined as

$$D(\tau) = \frac{1}{\sigma_\theta}\sqrt{\sum_{t=0}^{N-\tau} \frac{[\Theta(t+\tau) - \Theta(t) - \overline{\Theta(t+\tau) - \Theta(t)}]^2}{N-\tau-1}}\,, \tag{26}$$

where

$$\overline{\Theta(t+\tau) - \Theta(t)} = \sum_{t=0}^{N-\tau} \frac{\Theta(t+\tau) - \Theta(t)}{N-\tau}\,, \tag{27}$$

N is the number of data points and times t and τ are measured in cycle period units. It is easy to realize that Eq. (27) ensures that $D(\tau = 1) = 1$. In the presence of fractal statistics we would have, based on the discussion of anomalous diffusion in Section 1,

$$D(\tau) \propto \tau^{\beta} = \tau^{\alpha/2}\,. \tag{28}$$

Fig. 4 shows the SDA for the residual part of the KL decomposition. Each set of graphs concerning the same crack data look quite similar. All three sets of graphs show that the curves have a initial scaling exponent approximately within the range $0.5 < \beta < 0.9$. The mean curve value is represented by the curves with black circles in Fig. 4. These early time values of β, interpreted in terms of the random walks discussed in Section 1, indicate that the residual parts of the signal manifest a persistent behavior, i.e., a persistent correlation that lasts at least 10 consecutive cycles on average.

For $10 < \tau < 100$ the data present a slight antipersistency with $0.4 < H < 0.5$. Consequently, the residual process is initially strongly persistent, but asymptotically it is almost random. We observe that for $10 < \tau < 100$ the mean scaling exponent is approximately $H = 0.45$ in the case of the linear continuous model and this is slightly larger than the scaling exponent in the KL discrete decomposition. This change in scaling is due to the fact that the principal part obtained with the KL decomposition extracts more information from the original signal than does the simple linear approximation.

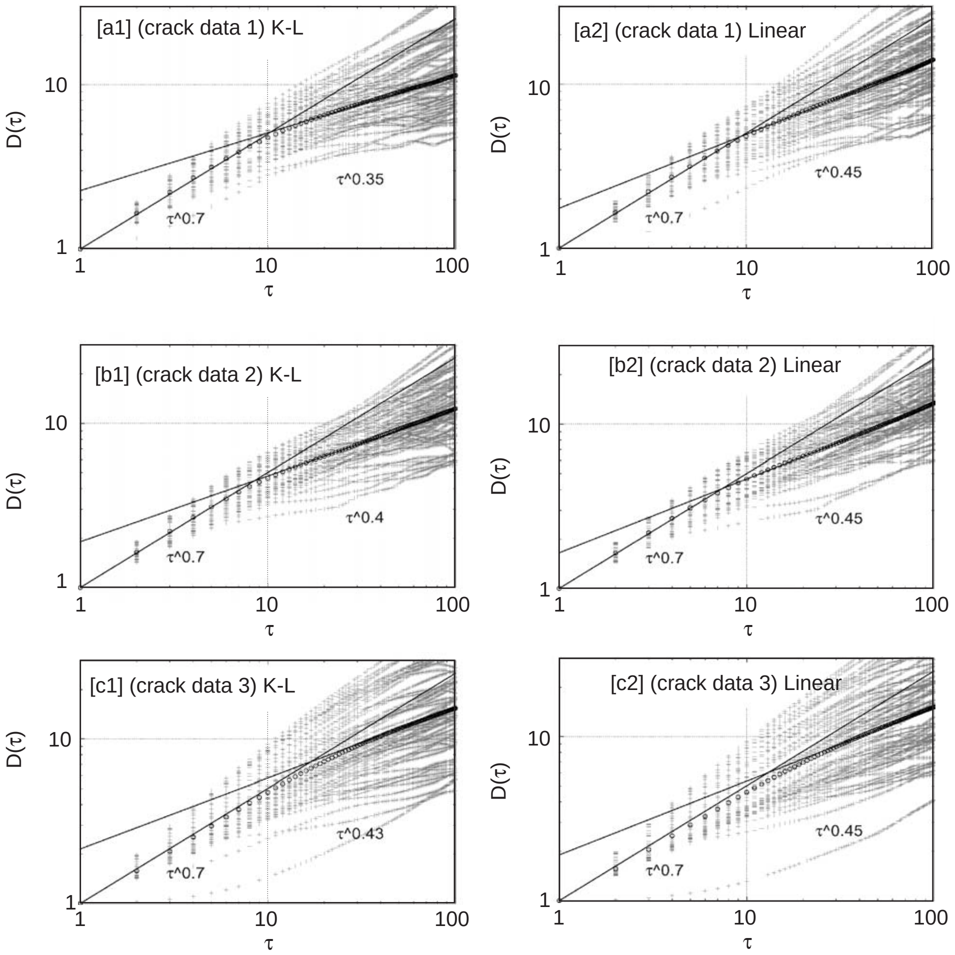


Fig. 4. SDA for the residual part of the KL decomposition (left figures) against SDA for the residual part of linear approximation (right figures) of the continuous model. Note the scaling transition at $\tau \approx 10$ from $H \approx 0.7$ to $H \approx 0.5$ in both cases for all data sets.

In Section 1 we have explained that an initial anomalous diffusion that lasts up to a certain τ as detected by Eq. (1) could also be an artifact related not to some autocorrelation pattern in the data but to the transition from the initial geometrical properties of the distribution of the events $\{\xi_i\}$ of a time series to the Gaussian shape of the asymptotic diffusion distribution. To check that the persistent behavior for $\tau < 10$ observed in the plots of Fig. 4 expresses real correlation patterns, we repeat SDA of the data after randomizing the time series of the increments $\{\theta_t\}$. That is, for each crack data first we have the sequence $\{\theta_t\}$ defined as $\theta_t = \Theta(t) - \Theta(t-1)$, then we shuffle $\{\theta_t\}$ and obtain a new sequence $\{\theta'_t\}$ and generate a new walk $\Theta'(t) = \sum_{i=1}^{t} \theta'_i$, and finally we apply SDA to the new curve $\Theta'(t)$. Fig. 5 shows

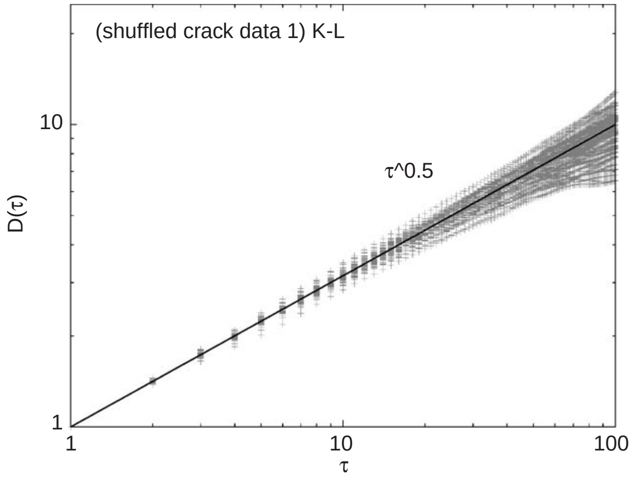


Fig. 5. SDA for the residual part of the KL decomposition after shuffling of the increments $\{\theta_t\}$. Note the random scaling of $H \approx 0.5$. The data refer to set #1 and the comparison has to be made with Fig. 4 (crack data 1) KL.

the result for the crack set #1 where the residual part is estimated with the KL decomposition; for the other data sets the results are similar. Fig. 5 clearly shows that after shuffling of the temporal order of the single increments $\{\theta_t\}$, the SDA of the new sequence gives a scaling value of approximately $H = 0.5$ and the persistent behavior for$\tau < 10$ observed in Fig. 4 is absent. Thus, we conclude that the persistent behavior for $\tau < 10$ observed in Fig. 4 expresses real correlation patterns in the fluctuations of crack growth.

Fig. 6 also shows that the distributions of the scaling exponent seems to be quite uniform in the interval $0.5 < \beta < 0.9$ (with a probability $P > 0.9\%$) or, perhaps, as Fig. 6c shows better, there might be a slight prominence or skewness in favor of small value of β. In any case, all figures show that the distribution of the scaling exponent for the residual components of the curve obtained with the KL decomposition or the linear component of the continuous model practically coincide for all three data sets. This equivalence suggests that the continuous linear model essentially captures not only the dominant properties of the signal, as obtained through the KL decomposition, see Eq. (24), but also the stochastic properties of the residual signal, as suggested in Eq. (25).

4.2. Statistics of damage measure estimates

We investigate the statistics of the damage measure estimates using a lognormal distribution. This is in keeping with the analysis of several investigators who assumed the crack growth rate in ductile alloys is lognormal-distributed (see, for example, the citations in Sobczyk and Spencer [9]). Other investigators have treated the crack length as being lognormal-distributed [37], rather than the residual

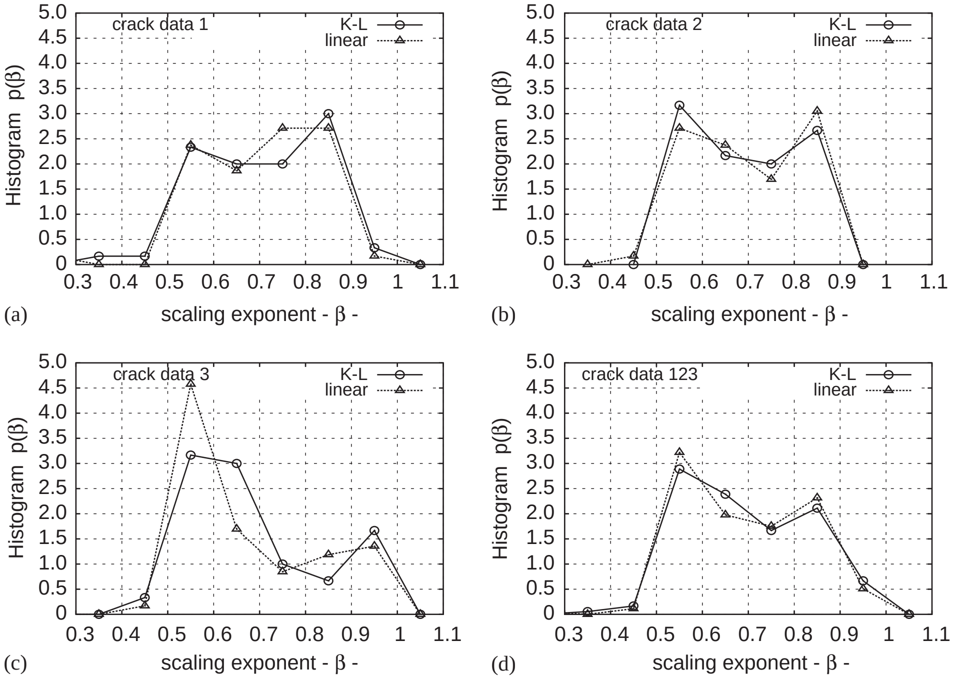


Fig. 6. Histogram and probability density function of the scaling exponent β estimated by fitting the range $1<\tau<=10$ for each curve shown in the plates of Fig. 4. Each plate compares the distributions obtained with the KL decomposition and the continuous linear model for each of the three crack data sets. Fig. 7d shows the histograms of all data.

fluctuations. The results of KL decomposition in Eqs. (16)–(19) are consistent with these assumptions because Ω_0, which dominates the random behavior of fatigue crack growth, can be considered to be a perfectly correlated (ballistic) random process, whereas the non-negative, multiplicative uncertainty term $\Theta(t, t_0)$ is a weakly (positively) correlated random process. Yang and Manning [39] have presented an empirical second-order approximation to crack growth by postulating a lognormal distribution of a parameter that does not bear any physical relationship to ΔS but is, to some extent, similar to $\Omega_0(\Delta S)$ in the present model.

Fig. 7 shows the histogram of the slopes $\Delta S^m \Omega_0$ of the curves according to the continuous model for the experimental data presented by Eq. (25), such as those shown in Fig. 3c. The histograms are fitted with the lognormal distribution $p(x, \mu, \sigma)$:

$$p(x; \mu, \sigma) = \frac{1}{x\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(\ln(x) - \mu)^2}{2\sigma^2}\right] . \tag{29}$$

The measured parameters μ and σ are recorded in Table 2. Finally, the parameters μ and σ are functions of $\mu_0 = \langle x \rangle$ and $\sigma_0^2 = \langle (x - \mu_0)^2 \rangle$ as follows:

$$\mu \equiv \ln(\mu_0) - \sigma^2/2 \tag{30}$$

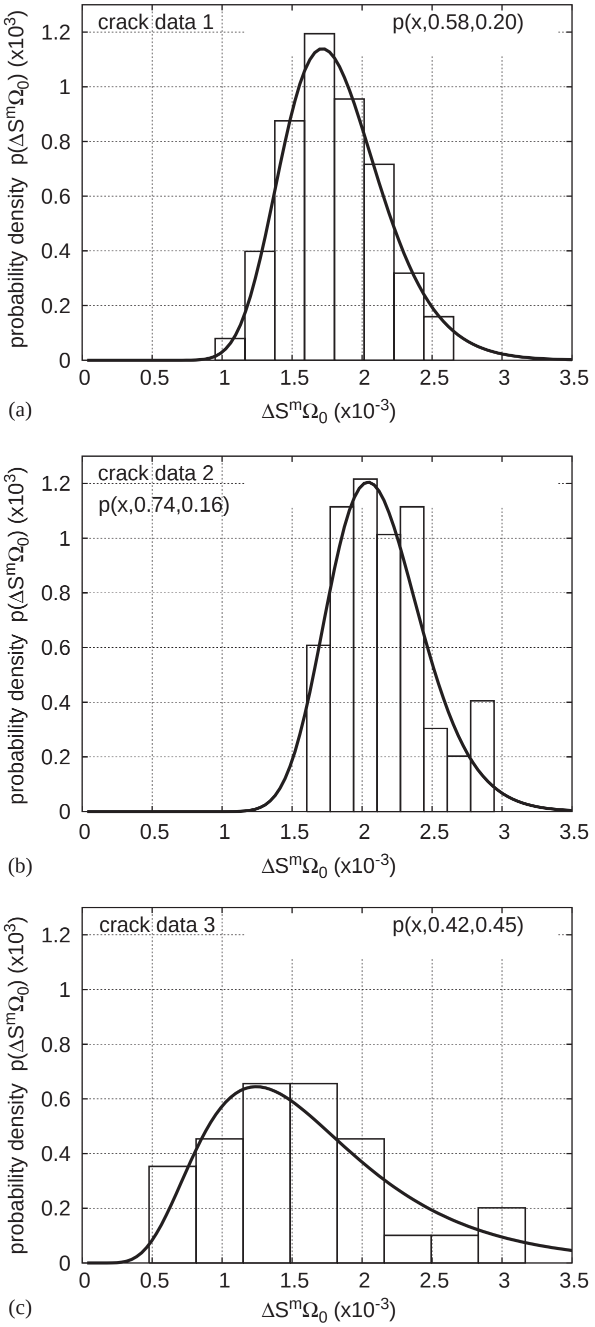


Fig. 7. Histogram of the quantities $\Delta S^m \Omega_0$ of the continuous model of the experimental data presented by Eq. (25). The histograms are fitted with a lognormal distribution $p(x; \mu, \sigma)$ shown in Eq. (29).

Table 2
Mean standard deviation of the increments of the residual part obtained with the KL decomposition and the residual part of the continuous model

	KL	Linear model
Set #1: $\langle\sigma_\theta\rangle =$	0.0024 ± 0.001	0.0025 ± 0.001
Set #2: $\langle\sigma_\theta\rangle =$	0.0024 ± 0.001	0.0025 ± 0.001
Set #3: $\langle\sigma_\theta\rangle =$	0.0038 ± 0.003	0.0043 ± 0.003

and

$$\sigma^2 \equiv \ln\left[1 + \left(\frac{\sigma_0}{\mu_0}\right)^2\right]. \tag{31}$$

Since the random parameter $\Delta S^m \Omega_0$ is not explicitly dependent on time, its expected value is obtained from Eq. (13) as

$$\mu_0 = \langle \Delta S^m \Omega_0 \rangle = \left\langle \frac{\psi(t, t_0)}{t - t_0} \right\rangle, \tag{32}$$

which is readily determined from the ensemble average estimate from each of the data sets. Asymptotically in time we find for the variance of $\Delta S^m \Omega_0$

$$\sigma_0^2 = \langle (\Delta S^m \Omega_0 - \mu_0)^2 \rangle = \left\langle \left[\frac{\psi(t, t_0)}{t - t_0}\right]^2 \right\rangle - \mu_0^2, \tag{33}$$

so that the variance can be determined directly from the ensemble average estimate from each of the data sets.

5. Crack model simulation

This section presents the results of Monte Carlo simulation of the fatigue crack damage process based on the model as it emerges from the stochastic analysis made in the previous section. The model that we introduce approximately reproduces the stochastic properties of both the ballistic or principal part of the fatigue crack growth and the associated fluctuations around it. The model consists in generating independently the fluctuation and the principal part of the fatigue crack damage in such a way that they are statistically equivalent to the correspondent observations and then combining them. The crack model simulation is based on four steps:

- *Principal part* or ballistic growth: We generate 60 values of $\Delta S^m \Omega_0$, lognormally distributed according to Eq. (29), where the parameters μ and σ are given by the actual fit of the phenomenological distribution shown in Fig. 6 and recorded in Table 2. A sample of the curves $\Delta S^m \Omega_0(t - t_0)$ simulating the data set #1 is shown in Fig. 8b.
- *Residual part* or fluctuations around the ballistic growth: We generate 60 fractal Gaussian noise sequences $\{\theta'_t\}$ each of length N of the original time sequence and

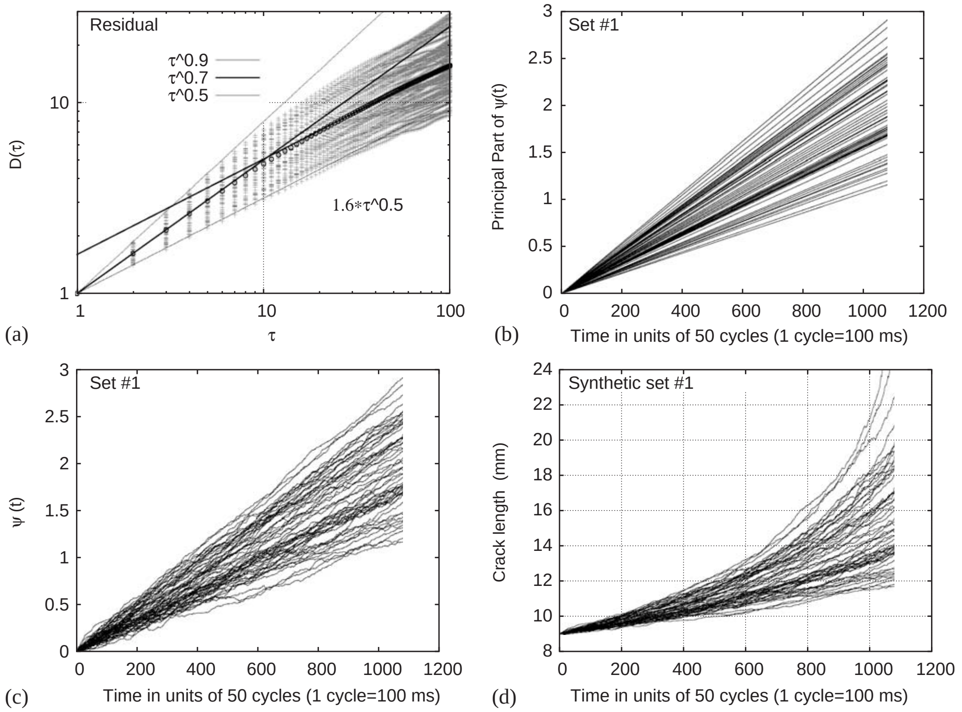


Fig. 8. Synthetic data of crack length for set #1. (a) SDA of the residual component; (b) ballistic growth; (c) damage increment $\psi(t)$; (d) crack length, compare with Fig. 1a.

with scaling exponent uniformly distributed in the interval $0.5<\beta<0.9$. The standard deviation of each sequence is set equal to the mean standard deviation of the increments of the residual component of the data reported in Table 1. To simulate the change of scaling exponent from persistent (for $\tau<10$) to almost random (for $\tau>10$), we section each fractal time series $\{\theta'_t\}$ into segments of length 10 within which the data would conserve the correlation, and finally we shuffle the position of these segments in the time series to reproduce a new time series $\{\theta_t\}$. These new time series will have persistent correlation for $\tau<10$ and uncorrelated randomness for $\tau>10$. Finally, the curve $\Theta(t)$ is obtained by integrating the new sequence $\{\theta_t\}$ and by detrending from it its linear component because the curve $\Theta(t)$ is supposed to have a zero mean. The SDA sample data analysis of an example of these synthetic residual data simulating the data set #1 is shown in Fig. 8a.

- The ballistic growth estimated in the principal part and the associated fluctuations of the residual part are combined according to Eq. (18) to obtain a simulated damage increment measure $\psi(t, t_0)$ for all 60 sequences and for the three data sets. Fig. 8c shows the simulated damage increment measure $\psi(t, t_0)$ simulating the data set #1.

- Finally, by using the respective value of the exponent m, reported in Section 4, for the data set #1 and a one-dimensional root-finding computer algorithm, Eq. (12) is inverted to obtain a simulated normalized crack length growth curves c_t, as seen in Fig. 8d. The similitude between Figs. 8d and 1a is noteworthy and the figures for the other data sets look qualitatively very similar; hence they are not presented in this paper.

6. Summary and conclusions

This paper presents a stochastic measure of fatigue crack damage. We have focused on the correlation properties of the fluctuations around fatigue crack growth in ductile alloys. The model of crack damage measure indicates that the fluctuations around fatigue crack growth present strong correlation patterns within short-time scales and are uncorrelated for larger time scales. These findings suggest that the random stochastic models adopted in the present literature for describing the crack growth dynamics should be augmented with short-time correlated stochastic models.

The damage measure is modeled as an anomalous diffusion process that is obtained as a continuous function of the current crack length and of the initial crack length. Perhaps, the randomness in the damage measure estimate accrues primarily from manufacturing uncertainties such as defects generated during machining operations because such macro-defects are expected to drive the ballistic growth of cracks. This randomness is captured by a single lognormal-distributed random variable. Instead, the resulting diffusion process of estimated fluctuations around the ballistic growth of fatigue cracks is probably due to the inhomogeneity in the structural material because it is primarily associated with the micro-structure of the material, and is represented by a non-stationary fractional Brownian motion model. This non-stationarity manifests itself in the two scaling exponents occurring at different scales. Specifically, we observe a clear transition in the standard deviation analysis from an early time slope representing a strong persistence, $\beta \approx 0.7$ lasting for approximately $\tau \approx 10$ to a different slope asymptotically in time representing randomness, $\beta \approx 0.5$. This transition occurring at $\tau \approx 10$ from a scaling regime to another indicates the scale at which a structure change of the ductile alloys occurs.

The constitutive equation of the damage measure is based on the physics of fracture mechanics and is validated by KL decomposition of fatigue test data for 7075-T6 aluminum alloys at different levels of (constant-amplitude) cyclic load. The damage estimate is statistically orthogonal to the resulting zero-mean estimation error in the Hilbert space $L_2(P)$ defined by the probability measure of the stochastic damage measure. As such, the damage estimate is often viewed as a best least-square linear estimate. However, we find that the KL decomposition is statistically equivalent to the linear approximation in the continuum model that can be then used to simulate the fatigue crack growth in ductile alloys.

Acknowledgements

The authors are grateful to Professor H. Ghonem of University of Rhode Island for providing the test data of random fatigue crack growth. The work reported in this paper has been supported in part by the Army Research Office under Grant no. DAAD190110640. The first author thanks the Army Research Office for the support under Grant DAAG5598D0002.

Appendix: A supporting lemma

Lemma. *Let $A(\zeta)$ and $B(\zeta)$ be second-order real random variables; $x(\zeta, t)$ and $y(\zeta, t)$ be zero-mean mean-square continuous (possibly non-separable) real random processes; and the real $g(t)$ be almost everywhere continuous on an interval Δ such that, for all $t \in \Delta$, the following conditions hold:*

(i) $A(\zeta) \overset{ms}{=} B(\zeta)$;
(ii) $\langle A(\zeta)x(\zeta, t)\rangle = 0$ *and* $\langle B(\zeta)y(\zeta, t)\rangle = 0$.

Then, the following mean-square identity:

$$A(\zeta)g(t) + x(\zeta, t) \overset{ms}{=} B(\zeta)g(t) + y(\zeta, t)$$

yields

$$\left.\begin{aligned} x(\zeta, t) &= y(\zeta, t) \\ \langle A(\zeta)y(\zeta, t)\rangle &= 0 \\ \langle B(\zeta)x(\zeta, t)\rangle &= 0 \end{aligned}\right\} \quad \forall t \in \Delta\ .$$

Proof. It follows from the above mean-square identity that

$$Var[\{A(\zeta) - B(\zeta)\}g(t) + \{x(\zeta, t) - y(\zeta, t)\}] = 0$$

which may be expanded to yield

$$\begin{aligned} &Var[A(\zeta) - B(\zeta)]g(t)^2 + Var[x(\zeta, t) - y(\zeta, t)] \\ &\quad + \langle\{A(\zeta) - B(\zeta)\}\{x(\zeta, t) - y(\zeta, t)\}\rangle g(t) = 0\ . \end{aligned}$$

A combination of condition (i) and Schwarz inequality yields

$$Var[x(\zeta, t) - y(\zeta, t)] = 0$$

and the remaining two identities follow from condition (ii). □

References

[1] J.L. Bogdanoff, F. Kozin, Probabilistic Models of Cumulative Damage, Wiley, New York, 1985.
[2] O. Ditlevsen, Eng. Fracture Mech. 23 (2) (1986) 467.

[3] Y.K. Lin, J.N. Yang, AIAA J. 23 (1) (1985) 117.
[4] B.F. Spencer, J. Tang, M.E. Artley, AIAA J. 27 (11) (1989) 1628.
[5] H. Ishikawa, A. Tsurui, H. Tanaka, H. Ishikawa, Prob. Eng. Mech. 8 (1993) 43.
[6] P.E. Kloeden, E. Platen, Numerical Solution of Stochastic Differential Equations, Springer, Berlin, 1995.
[7] A. Tsurui, H. Ishikawa, Structural Safety 4 (1986) 15.
[8] V.V. Bolotin, Prediction of Service Life for Machines and Structures, ASME Press, New York, 1989.
[9] K. Sobczyk, B.F. Spencer, Random Fatigue: Data to Theory, Academic Press, Boston, MA, 1992.
[10] F. Casciati, P. Colombi, L. Farvelli, Fatigue Fract. Eng. Mater. Struct. 15 (5) (1992) 463.
[11] B.V. Chirikov, Phys. Rep. 52 (1979) 265.
[12] S. Havlin, D. Ben-Hvraham, Adv. Phys. 36 (1987) 695.
[13] N. Scafetta, P. Grigolini, Phys. Rev. E 66 (2002) 036130.
[14] B.J. West, Physiology, Promiscuity and Prophecy at the Millennium: A Tale of Tails, World Scientific, Singapore, 1999.
[15] B. Mandelbrot, J.W. Van Ness, SIAM Rev. 10 (1968) 422.
[16] M.F. Shlesinger, B.J. West, J. Klafter, Phys. Rev. Lett. 58 (1987) 1100.
[17] L.F. Richardson, Proc. R. Soc. London Ser. A 110 (1926) 709.
[18] B. Gutenberg, C.F. Richter, Bull. Seismol. Soc. 34 (1994) 185;
J.M. Carlson, J.S. Langer, B.E. Shaw, Rev. Mod. Phys. 66 (1994) 657.
[19] O. Peters, C. Hertlein, K. Christensen, Phys. Rev. Lett. 88 (2002) 018701.
[20] U. Frisch, Turbulence: The Legacy of A.N. Kolmogorov, Cambridge University Press, Cambridge USA, 1992.
[21] W.G. Glöckle, T.F. Nonnenmacher, J. Stat. Phys. 71 (1993) 741;
W.G. Glöckle, T.F. Nonnenmacher, Rheol. Acta. 33 (1994) 337.
[22] P. Grigolini, D. Leddon, N. Scafetta, Phys. Rev. E 65 (2002) 046203.
[23] N. Scafetta, B.J. West, Phys. Rev. Lett. 90 (2003) 248701.
[24] N. Scafetta, P. Grigolini, T. Imholt, J.A. Roberts, B.J. West, Phys. Rev. E 69 (2004) 026303.
[25] N. Scafetta, B.J. West, Phys. Rev. Lett. 92 (2004) 138501.
[26] N. Scafetta, V. Latora, P. Grigolini, Phys. Rev. E 66 (2002) 031906.
[27] P.C. Paris, F. Erdogan, J. Basic Eng. Trans. ASME, D 85 (1963) 528.
[28] J. Schijve, Fatigue Crack Growth Under Spectrum Loads, ASTM STP 595 (1976) 3.
[29] A. Ray, R. Patankar, Eng. Fracture Mech. 62 (1999) 477.
[30] W. Elber, Eng. Fracture Mech. 2 (1970) 37.
[31] T.R. Porter, Eng. Fracture Mech. 4 (1972) 717.
[32] J.C. McMillian, R.M.N. Pelloux, ASTM STP 748 (1981) 53–84.
[33] H. Ghonem, S. Dore, Eng. Fracture Mech. 27 (1987) 1.
[34] Y.N. Rabotnov, Elements of Hereditary Solid Mechanics, MIT Publishers, Moscow, 1980.
[35] T.L. Anderson, Fracture Mechanics, second ed., CRC Press, Boca Raton, FL, 1995.
[36] J.A. Bannantine, J.J. Comer, J.L. Handrock, Fundamentals of Metal Fatigue Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1990.
[37] A. Ray, ASME J. Dyn. Syst. Meas. Cont. 121 (1999) 386.
[38] A.H. Jazwinski, Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.
[39] E. Wong, B. Hajek, Stochastic Processes in Engineering Systems, Springer, New York, 1985.
[40] K. Fukunaga, Introduction to Statistical Pattern Recognition, second ed., Academic Press, Boston, 1990.

International Journal of Control
Vol. 78, No. 12, 15 August 2005, 949–967

Taylor & Francis
Taylor & Francis Group

Signed real measure of regular languages for discrete event supervisory control

A. RAY*

Department of Mechanical Engineering, The Pennsylvania State University, University Park, PA 16802, USA

(*Received 9 October 2004; in final form 27 May 2005*)

This paper reviews, expands, and clarifies the underlying concepts of a signed real measure of regular languages, which has been used as a novel tool for synthesis of discrete event supervisory control systems. The language measure is constructed upon the principles of automata theory and real analysis. It allows total ordering of a set of partially ordered sublanguages of a regular language for quantitative evaluation of the supervised behaviour of deterministic finite state automata (DFSA) under different supervisors. In the setting of the language measure, a supervisor's performance is superior if the supervised plant is more likely to terminate at a good marked state and/or less likely to terminate at a bad marked state. The computational complexity of the language measure algorithm is polynomial in the number of DFSA states.

1. Introduction

Discrete event systems belong to a special class of dynamical systems. The states of a discrete event system may take discrete (or symbolic) values and change only at (possibly asynchronous) discrete instants of time, in contrast to the familiar continuously varying dynamical systems of the physical world, which can be modelled by differential or difference equations. The dynamics of many human-engineered systems evolve asynchronously in time via complex interactions of various discrete-valued events with continuously varying physical processes. The relatively young discipline of discrete event systems has undergone rapid growth over the last three decades with the evolution of human engineered complex systems, such as integrated control and communication systems, distributed sensing and monitoring of large-scale engineering systems, manufacturing and production systems, software fault management, and military Command, Control, Computer, Communication, Intelligence, Surveillance, and Reconnaissance (C^4ISR) systems.

The discipline of discrete event systems was initiated with simulation of human-engineered processes about four decades ago in the middle of nineteen sixties. The art of discrete event simulation emerged with the development of a simulation software package, called GPSS, that was followed by numerous other software simulation tools, such as SIMSCRIPT II.5, SLAM II, and SIMAN (Law and Kelton 1991). Shortly thereafter, computer scientists and control theorists entered the field and brought in theoretical concepts of languages and automata in modelling discrete event systems. In the late nineteen sixties, Arbib (see Kalman *et al.* 1969) showed how algebraic methods could be used to explore the structure of finite automata to model dynamical systems. Around that time, computer scientists focused on formal languages, automata theory, and computational complexity for application of language-theoretic concepts (e.g., regular expressions and context-free grammars) in software development including design of compilers and text processors (Yu 1997, Hopcroft *et al.* 2001). In the late nineteen seventies and early nineteen eighties, Ho and co-workers introduced the concept of finite perturbation in discrete event systems for modelling and analysis of human-engineered systems (Ho and Cao 1991). So far, no concrete theoretical concept and mathematical tools had been available for analysis and synthesis of discrete event control systems.

The concept of discrete event supervisory (DES) control was first introduced in the seminal paper of

*Email: axr2@psu.edu

International Journal of Control
ISSN 0020–7179 print/ISSN 1366–5820 online © 2005 Taylor & Francis Group Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/00207170500202447

Ramadge and Wonham (1987) and this important paradigm has been subsequently extended by other researchers (for example, see citations in Kumar and Garg (1995) and Cassandras and Lafortune (1999), and the October 2000 issue of Part B of IEEE Transactions on Systems, Man, and Cybernetics). These efforts have led to the evolution of a new discipline in decision and Control, called Supervisory Control Theory (SCT), that requires partitioning the discrete-event behaviour of a physical process, called the plant, into legal and illegal categories. The legal behaviour of plant dynamics is modelled by a deterministic finite-state automaton, abbreviated as DFSA in the sequel. The DFSA model is equivalent to a regular language that is built upon an alphabet of finitely many events; the event alphabet is partitioned into subsets of controllable events (that can be disabled) and uncontrollable events (that cannot be disabled). Based on the regular language of an unsupervised plant, SCT synthesizes a DES controller as another regular language, having the common alphabet with the plant language, that guarantees restricted legal behaviour of the supervised plant based on the desired specifications. Instead of continuously handling numerical data, DES controllers are designed to process event strings to disable certain controllable events in the physical plant. A number of algorithms for DES control synthesis have evolved based on the automata theory and formal languages relying on the disciplines of Computer Science and Control Science. In general, a supervised plant DFSA is synthesized as a parallel composition of the unsupervised plant DFSA and a supervisor DFSA (Cassandras and Lafortune 1999). The supervised plant DFSA yields a sublanguage of the unsupervised plant language, which enables restricted legal behaviour of the supervised plant (Ramadge and Wonham 1987, Kumar and Garg 1995, Cassandras and Lafortune 1999). These concepts have been extended to several practical applications, including hierarchical Command, Control, Communication, and Intelligence (C^3I) systems (Phoha *et al.* 2002). Apparently, there have been no quantitative methods for evaluating the performance of supervisory controllers and establishing thresholds for their performance.

The concept of permissiveness has been used in DES control literature (Kumar and Garg 1995, Cassandras and Lafortune 1999) to facilitate qualitative comparison of DES controllers under the language controllability condition. Design of maximally permissive DES controllers has been proposed by several researchers based on different assumptions. However, maximal permissiveness does not imply best performance of the supervised plant from the perspective of achieving plant operational objectives. For example, in the travelling salesman problem, a maximally permissive supervisor may not yield the least expensive way of visiting the scheduled cities and returning to the starting point because no quantitative measure of performance is addressed in this type of supervisor design.

The above argument evinces the need for a signed real measure of regular languages, which can be used for quantitative evaluation and comparison of different supervisors for a physical plant, instead of relying on permissiveness as the (qualitative) performance index. Construction of the proposed language measure follows Myhill–Nerode Theorem (Martin 1997, Hopcroft *et al.* 2001), which states that a regular language can be partitioned into finitely many right-invariant equivalence classes. In other words, a state-based partitioning of the (unsupervised) plant language yields equivalence classes of finite-length event strings. Each marked state is characterized by a signed real value that is chosen based on the designer's perception of the state's impact on the system performance. Conceptually similar to conditional probability, each event is assigned a cost based on the state at which it is generated. This procedure permits a string of events, terminating on a good (bad) marked state, to have a positive (negative) measure. A supervisor can be designed in this setting such that the supervisor attempts to eliminate as many bad strings as possible and retain as many good strings as possible. Different supervisors may achieve this goal in different ways and generate a partially ordered set of supervised sublanguages. The language measure then creates a total ordering on the performance of the supervised sublanguages, which provides a precise quantitative comparison of the controlled plant behaviour under different supervisors. This feature is formally stated as follows.

> *Given that the relation $\subseteq$ induces a partial ordering on a set of supervised sublanguages $\{L(S^j/G), j = 1, \ldots, N\}$ of the plant language $L(G)$ under supervisors whose languages are $\{L(S^j), j = 1, \ldots, N\}$, the language measure μ induces a total ordering $\leq$ on $\{\mu(L(S^j/G))\}$. In other words, the range of the set function μ is totally ordered while its domain could be partially ordered.*

The above problem was first addressed by Wang and Ray (2004) who proposed a signed measure of regular languages; an alternative approach was proposed by Ray and Phoha (2003) who constructed a vector space of formal languages and defined a metric based on the total variation measure of the language.

This paper reviews, clarifies and expands the contents of previous publications (Ray and Phoha 2003, Wang and Ray 2004) from the perspectives of discrete-event supervisory control within a unified framework and also introduces new concepts and ramifications of the language measure and its parameter identification. Systematic procedures for computation of the language measure are developed in this paper and they are illustrated with an engineering example. The major objective here is rigorous formulation and systematic construction of a real signed measure of regular languages, based on the fundamental principles of automata theory and real analysis. The quantitative tools are readily applicable to analysis and synthesis of discrete-event supervisory control algorithms. Specifically, performance indices of supervisors can be defined in terms of the language measure.

The signed real measure for a DFSA, presented in this paper, is constructed based on assignment of an event cost matrix and a characteristic vector. Two techniques for language measure computation have been recently reported. While the first technique (Wang and Ray 2004) leads to a system of linear equations whose (closed form) solution yields the language measure vector, the second technique (Ray and Phoha 2003) is a recursive procedure with finite iterations. A sufficient condition for finiteness of the signed measure has been established in both cases; and an upper bound is established for the max norm of the language measure vector.

In order to induce total ordering on the measure of different sublanguages of a plant language under different supervisors, it is implicit that same strings in different sublanguages must be assigned the same measure. This is accomplished by a quantitative tool that requires a systematic procedure to assign a characteristic vector and an event cost matrix. The clarifications and extensions presented in this paper are intended to enhance development of a systematic analytical tool for synthesizing discrete-event supervisory control. For example, Ray *et al.* (2004) have proposed unconstrained optimal control of regular languages where a state-based optimal control policy is obtained by selectively disabling controllable events to maximize the measure of the supervised plant language.

The paper is organized in eight sections including the present introductory section and two appendices. Section 2 briefly describes the language measure and introduces the notations. Section 3 presents the procedure by which the performance of different supervisors can be compared based on a common quantitative tool. It also discusses two methods for computing language measure. Section 4 addresses issues regarding physical interpretation of the event cost used in the language measure. Section 5 presents a recursive algorithm for identification of the language parameters (i.e., elements of the event cost matrix). Section 6 illustrates the usage of the language measure for construction of metric spaces of formal languages and synthesis of optimal discrete-event supervisors. Section 7 presents an application of the language measure on the discrete-event model of a twin-engine unmanned aircraft (Ray and Phoha 2003, Ray *et al.* 2004). The paper is summarized and concluded in §8 along with recommendations for future research. Appendix I provides pertinent mathematical background of measure theory as needed in the main body of the paper. Appendix II establishes a sufficient condition for absolute convergence of the language measure.

2. Language measure concept

This section first introduces the signed real measure of regular languages, originally reported in (Ray and Phoha 2003, Wang and Ray 2004). Then, the underlying concepts of language measure are clarified in the context of discrete event supervisory (DES) control.

Let $G_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$ be a trim (i.e., accessible and co-accessible) finite-state automaton model (Ramadge and Wonham 1987, Cassandras and Lafortune 1999) that represents the discrete-event dynamics of a physical plant, where $Q = \{q_k \colon k \in \mathcal{I}_Q\}$ is the set of states and $\mathcal{I}_Q \equiv \{1, 2, \ldots, n\}$ is the index set of states; the automaton starts with the initial state q_i; the alphabet of events is $\Sigma = \{\sigma_k \colon k \in \mathcal{I}_\Sigma\}$, and $\mathcal{I}_\Sigma \equiv \{1, 2, \ldots, \ell\}$ is the index set of events; $\delta \colon Q \times \Sigma \rightarrow Q$ is the (possibly partial) function of state transitions; and $Q_m \equiv \{q_{m_1}, q_{m_2}, \ldots, q_{m_r}\} \subseteq Q$ is the set of marked (i.e., accepted) states with $q_{m_k} = q_j$ for some $j \in \mathcal{I}_Q$.

Let Σ^* be the Kleene closure of Σ, i.e., the set of all finite-length strings made of the events belonging to Σ as well as the empty string ϵ that is viewed as the identity of the monoid Σ^* under the operation of string concatenation, i.e., $\epsilon s = s = s\epsilon$. The extension $\hat{\delta} \colon Q \times \Sigma^* \rightarrow Q$ is defined recursively in the usual sense (Martin 1997, Hopcroft *et al.* 2001). For DES control (Ramadge and Wonham 1987), the event alphabet Σ is partitioned into sets, Σ_c and $\Sigma - \Sigma_c$ of controllable and uncontrollable events, respectively, where each event in Σ_c and no event in $\Sigma - \Sigma_c$ can be disabled by the supervisor.

Definition 1: The language $L(G_i)$ generated by a DFSA G_i initialized at the state $q_i \in Q$ is defined as

$$L(G_i) = \{s \in \Sigma^* \mid \hat{\delta}(q_i, s) \in Q\}. \tag{1}$$

Since the state transition function δ is allowed to be a partial function, $L(G_i) \subseteq \Sigma^*$ following Definition 1; if δ is a total function, then the generated language $L(G_i) = \Sigma^*$.

Definition 2: Given a DFSA plant model G_i, having the set of controllable events $\Sigma_c \subseteq \Sigma$, let S and $\widetilde{S}$ be two controllable supervisors (i.e., each of S and $\widetilde{S}$ is represented by an event disabling mapping $L(G_i) \rightarrow 2^{\Sigma_c}$). Let the languages of the plant supervised by S and $\widetilde{S}$

be denoted as $L(S/G_i)$ and $L(\widetilde{S}/G_i)$, respectively. Then, S is said to be less permissive (or more restrictive) than $\widetilde{S}$, denoted as $S \preceq \widetilde{S}$, if the following condition holds.

$$S \preceq \widetilde{S} \quad \text{if } L(S/G_i) \subseteq L(\widetilde{S}/G_i). \tag{2}$$

In other words, S may disable a larger set of controllable events than $\widetilde{S}$ following the execution of an event string $s \in \Sigma^*$.

Definition 3: The language $L_m(G_i)$ marked by a DFSA G_i, initialized at the state $q_i \in Q$, is defined as

$$L_m(G_i) = \{s \in \Sigma^* \mid \hat{\delta}(q_i, s) \in Q_m\}. \tag{3}$$

Definition 4: For every $q_i, q_k \in Q$, let $L_{i,k}$ denote the set of all strings that, starting from the state q_i, terminate at the state q_k, i.e.,

$$L_{i,k} = \{s \in \Sigma^* \mid \hat{\delta}(q_i, s) = q_k\}. \tag{4}$$

In order to obtain a quantitative measure of the marked language, the set Q_m of marked states is partitioned into Q_m^+ and Q_m^-, i.e., $Q_m = Q_m^+ \cup Q_m^-$ and $Q_m^+ \cap Q_m^- = \emptyset$. The positive set Q_m^+ contains all good marked states that one would desire to reach, and the negative set Q_m^- contains all bad marked states that one would not want to terminate on, although it may not always be possible to completely avoid the bad states while attempting to reach the good states. From this perspective, each marked state is characterized by an assigned real value that is chosen based on the designer's perception of the state's impact on the system performance.

Definition 5: The characteristic function $\chi\colon Q \to [-1, 1]$ assigns a signed real weight to a state-based sublanguage $L_{i,j}$, having each of its strings terminating on the same state q_j, and is defined as

$$\forall q_j \in Q, \quad \chi(q_j) \in \begin{cases} [-1, 0), & q_j \in Q_m^- \\ \{0\}, & q_j \notin Q_m \\ (0, 1], & q_j \in Q_m^+. \end{cases} \tag{5}$$

The state weighting vector, denoted by $\mathbf{X} = [\chi_1\ \chi_2\ \ldots\ \chi_n]^T$, is called the $\mathbf{X}$-vector, where $\chi_j \equiv \chi(q_j)$. That is, the jth element χ_j of $\mathbf{X}$-vector is the weight assigned to the corresponding state q_j.

In general, the marked language $L_m(G_i)$ consists of both good and bad strings, which start from the initial state q_i, respectively lead to Q_m^+ and Q_m^-. Denoting the set difference operation by "–", any event string belonging to the language $L^0(G_i) \equiv L(G_i) - L_m(G_i)$ leads to one of the non-marked states belonging to $Q - Q_m$ and $L^0(G_i)$ does not contain any one of the good or bad strings. Partitioning Q_m into the positive set Q_m^+ and the negative set Q_m^- leads to partitioning of the marked language $L_m(G_i)$ into a positive language $L_m^+(G_i)$ and a negative language $L_m^-(G_i)$. Based on the equivalence classes defined in the Myhill–Nerode Theorem (Hopcroft *et al.* 2001), the regular languages $L(G_i)$ and $L_m(G_i)$ can be expressed as

$$L(G_i) = \bigcup_{k \in \mathcal{I}_Q} L_{i,k} \tag{6}$$

$$L_m(G_i) = L_m^+(G_i) \bigcup L_m^-(G_i) \tag{7}$$

where the sublanguage $L_{i,k} \subseteq L(G_i)$ is uniquely labelled by the state q_k, $k \in \mathcal{I}_Q$ and $L_{i,k} \cap L_{i,j} = \emptyset \ \ \forall k \neq j$; and $L_m^+(G_i) \equiv \cup_{q_k \in Q_m^+} L_{i,k}$ and $L_m^-(G_i) \equiv \cup_{q_k \in Q_m^-} L_{i,k}$ are good and bad sublanguages of $L_m(G_i)$, respectively. Then, the null sublanguage $L^0(G_i) = \cup_{q_k \notin Q_m} L_{i,k}$ and $L(G_i) = L^0(G_i) \cup L_m^+(G_i) \cup L_m^-(G_i)$.

Now a signed real measure is constructed as $\mu^i\colon 2^{L(G_i)} \to \mathbf{R} \equiv (-\infty, \infty)$ on the σ-algebra $M = 2^{L(G_i)}$. (Appendix I provides details of measure-theoretic definitions and results.) With this choice of σ-algebra, every singleton set made of an event string $s \in L(G_i)$ is a measurable set, which allows its quantitative evaluation based on the above state-based decomposition of $L(G_i)$ into null (i.e., $L^0(G_i)$), positive (i.e., $L_m^+(G_i)$), and negative (i.e., $L_m^-(G_i)$) sublanguages.

Conceptually similar to the conditional probability, each event is assigned a cost based on the state at which it is generated.

Definition 6: The event cost of the DFSA G_i is defined as a (possibly partial) function $\tilde{\pi}\colon \Sigma^* \times Q \to [0, 1]$ such that $\forall q_i \in Q$, $\forall \sigma_j \in \Sigma$, $\forall s \in \Sigma^*$,

$$\begin{aligned} &\tilde{\pi}[\sigma_j, q_i] = 0 \text{ if } \delta(q_i, \sigma_j) \text{ is undefined; } \tilde{\pi}[\epsilon, q_i] = 1; \\ &\tilde{\pi}[\sigma_j, q_i] \equiv \tilde{\pi}_{ij} \in [0, 1); \ \sum_{j \in \mathcal{I}_\Sigma} \tilde{\pi}_{ij} < 1; \\ &\tilde{\pi}[\sigma_j s, q_i] = \tilde{\pi}[\sigma_j, q_i]\, \tilde{\pi}[s, \delta(q_i, \sigma_j)]. \end{aligned} \tag{8}$$

A simple application of the induction principle to the last part of Definition 6 shows $\tilde{\pi}[st, q_j] = \tilde{\pi}[s, q_j]\, \tilde{\pi}[t, \hat{\delta}(q_j, s)]$. The condition $\sum_{k \in \mathcal{I}_Q} \tilde{\pi}_{jk} < 1$ provides a sufficient condition for the existence of the real signed measure as discussed in § 3 and Appendix II. Additional comments on the physical interpretation of the event cost are provided in § 4.

The $n \times \ell$ event cost matrix is defined as

$$\widetilde{\mathbf{\Pi}} = \begin{bmatrix} \tilde{\pi}_{11} & \tilde{\pi}_{12} & \ldots & \tilde{\pi}_{1\ell} \\ \tilde{\pi}_{21} & \tilde{\pi}_{22} & \ldots & \tilde{\pi}_{2\ell} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\pi}_{n1} & \tilde{\pi}_{n2} & \ldots & \tilde{\pi}_{n\ell} \end{bmatrix}. \tag{9}$$

Definition 7: The state transition cost, $\pi\colon Q \times Q \to [0,1)$, of the DFSA G_i is defined as follows.

$\forall i, j \in \mathcal{I}_Q$,

$$\pi_{ij} = \begin{cases} \sum_{\sigma \in \Sigma} \tilde{\pi}[\sigma, q_i], & \text{if } \delta(q_i, \sigma) = q_j \\ 0 & \text{if } \{\delta(q_i, \sigma) = q_j\} = \emptyset. \end{cases} \tag{10}$$

The $n \times n$ state transition cost matrix is defined as

$$\mathbf{\Pi} = \begin{bmatrix} \pi_{11} & \pi_{12} & \dots & \pi_{1n} \\ \pi_{21} & \pi_{22} & \dots & \pi_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{n1} & \pi_{n2} & \dots & \pi_{nn} \end{bmatrix} \tag{11}$$

and is referred to as the $\mathbf{\Pi}$-matrix in the sequel.

Definition 8: Given a DFSA $G_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$ the cost ν^i of a sublanguage $K \subseteq L(G_i)$ is defined as the sum of the event cost $\tilde{\pi}$ of individual strings belonging to K.

$$\nu^i(K) = \sum_{s \in K} \tilde{\pi}[s, q_i]. \tag{12}$$

Definition 9: For a given DFSA G_i, the signed real measure of every singleton string set $\{s\} \in L_{i,j} \subseteq L(G_i)$ is defined as $\mu^i(\{s\}) \equiv \tilde{\pi}(s, q_i)\chi_j$ implying that

$$\forall s \in L_{i,j}, \quad \mu^i(\{s\}) \begin{cases} = 0, & q_j \notin Q_m \\ > 0, & q_j \in Q_m^+ \\ < 0, & q_j \in Q_m^- \end{cases} \tag{13}$$

Thus an event string terminating on a good (bad) marked state has a positive (negative) measure and one terminating on a non-marked state has zero measure. It follows from Definition 9 that the signed measure of the sublanguage $L_{i,j} \subseteq L(G_i)$ of all events, starting at q_i and terminating at q_j, is

$$\mu^i(L_{i,j}) = \left(\sum_{s \in L_{i,j}} \tilde{\pi}[s, q_i] \right) \chi_j \tag{14}$$

Definition 10: The signed real measure of the language of a DFSA G_i initialized at a state $q_i \in Q$, is defined as

$$\mu_i \equiv \mu^i(L(G_i)) = \sum_{j \in \mathcal{I}_Q} \mu^i(L_{i,j}). \tag{15}$$

The language measure vector, denoted as $\boldsymbol{\mu} = [\mu_1 \ \mu_2 \dots \mu_n]^T$, is called the $\boldsymbol{\mu}$-vector.

Remark 1: $\mu^i(L_m(G_i)) = \mu_i \ \forall i \in \mathcal{I}_Q$ because $\chi_k = 0$ $\forall q_k \in Q - Q_m$.

It follows from Definition 10 that $\mu^i(L_{i,j}) = \nu^i(L_{i,j})\chi_j$. Under the condition of $\sum_k \tilde{\pi}_{jk} < 1$ in Definition 6, convergence of the signed real language measure μ^i has been proved in (Ray and Phoha 2003, Wang and Ray 2004). The total variation measure $|\mu^i|$ of μ^i has also been shown to be finite for every $i \in \mathcal{I}_Q$ (Ray and Phoha 2003).

In the above setting, the role of the language measure in DES control synthesis is explained below.

> *A discrete-event non-marking supervisor S restricts the marked behaviour of an unsupervised (i.e., uncontrolled) plant G_i such that $L_m(S/G_i) \subseteq L_m(G_i)$. The unsupervised marked language $L_m(G_i)$ consists of good strings leading to Q_m^+ and bad strings leading to Q_m^-. A supervised (i.e., controlled) language $L_m(S/G_i)$ based on a given specification of the supervisor S may disable some of the bad strings and keep some of the good strings enabled. Different supervisors $S_j\colon j \in \{1, 2, \dots, n_s\}$ for a DFSA G_i achieve this goal in different ways and generate a partially ordered set of supervised sublanguages $\{L_m(S_j/G_i)\colon j \in \{1, 2, \dots, n_s\}\}$. The real signed measure μ^i provides a precise quantitative comparison of the controlled plant behaviour under different supervisors because the set $\{\mu^i(L_m(S_j/G_i))\colon j \in \{1, 2, \dots, n_s\}\}$ is totally ordered.*

In order to realize the above goal, the performance of different supervisors has to be evaluated based on a common quantitative tool. Let $G \equiv \langle Q^G, \Sigma, \delta^G, q_1^G, Q_m^G \rangle$ denote the unsupervised plant and $S \equiv \langle Q^S, \Sigma, \delta^S, q_1^S, Q_m^S \rangle$ denote the supervisor with respective languages $L(G)$ and $L(S)$ and the corresponding marked languages $L_m(G)$ and $L_m(S)$.

Let $\mathbb{G} \equiv \langle Q, \Sigma, \delta, q_1, Q_m \rangle$ where $Q = Q^G \times Q^S$, $q_1 = (q_1^G, q_1^S)$, $Q_m = \{(p, \tilde{p}) | p \in Q_m^G$ and $\tilde{p} \in Q_m^S\}$ and the transition function δ is defined by the formula: $\forall p \in Q^G$, $\tilde{p} \in Q^S$, and $\sigma \in \Sigma$

$$\delta((p, \tilde{p}), \sigma) = \big(\delta^G(p, \sigma), \delta^S(\tilde{p}, \sigma)\big). \tag{16}$$

Then, the marked language $L_m(\mathbb{G})$ of the automaton $\mathbb{G}$ is $L_m(G) \cap L_m(S)$ because $\mathbb{G}$ is a parallel composition (Ramadge and Wonham 1987, Cassandras and Lafortune 1999) of the automata G and S that have the common alphabet Σ. Then, it follows that the extension $\hat{\delta}$ satisfies the condition

$$\forall s \in \Sigma^*, \ \hat{\delta}((p, \tilde{p}), s) = (\hat{\delta}^G(p, s), \hat{\delta}^S(\tilde{p}, s)) \tag{17}$$

whenever $\hat{\delta}^G(p, s)$ and $\hat{\delta}^S(\tilde{p}, s)$ are defined.

The unsupervised plant language $L(G)$ is partitioned by $L_{1,j}^G$, $1 \le j \le n^G$ where $|Q^G| = n^G$. Similarly, the supervisor language $L(S)$ is partitioned by $L_{1,k}^S$, $1 \le k \le n^S$ where $|Q^S| = n^S$. With this construction, each of the sublanguages $L_{1,j}^G$ is further partitioned by $L_{1,j}^G \cap L_{1,k}^S$. Thus, for any $q_j^G \in Q_m^G$, the set of strings, which is retained in $L_m(G) \cap L_m(S)$, is given by $L_{1,j}^G \cap (\cup_{q_k^S \in Q_m^S} L_{1,k}^S)$. In this setting, the goal is to synthesize a supervisor that will retain many strings that terminate on some state in Q_m^{G+} while discarding many strings that terminate on Q_m^{G-}. It will yield a relatively high measure and hence good performance.

The above construction shows how the event cost and characteristic function assigned to the unsupervised plant can be used as a quantitative tool with which the performance of different supervisors can be evaluated and compared. The following procedure indicates how this can be accomplished explicitly.

Definition 11: Let G, S and $\mathbb{G}$ be defined as above. Let G represent the unsupervised plant and $\tilde{\pi}^G$ be the event cost function and χ^G be the characteristic function. Then, for the DFSA $\mathbb{G}$ which represents the language of the supervised plant, the event cost function $\tilde{\pi}$ is defined as

$$\tilde{\pi}\Big[\sigma, \Big(q_i^G, q_j^S\Big)\Big] = \tilde{\pi}^G\big[\sigma, q_i^G\big]$$
$$\forall \sigma \in \Sigma \text{ and } \forall i, j \text{ s.t. } 1 \le i \le n^G, 1 \le j \le n^S. \tag{18}$$

The χ-vector for the DFSA $\mathbb{G}$ is defined as

$$\chi\Big(\Big(q_i^G, q_j^S\Big)\Big) = \chi^G(q_i^G)\mathcal{J}\Big(q_j^S\Big) \tag{19}$$

where $\mathcal{J}(\cdot)$ is the indicator function defined as

$$\mathcal{J}(p) = \begin{cases} 1 & p \in Q_m^S \\ 0 & p \notin Q_m^S. \end{cases} \tag{20}$$

Let $s \in L_{((q_1^G, q_1^S), (q_j^G, q_k^S))}$, i.e., the set of all strings starting at the state $(q_1^G, q_1^S) \in Q \equiv Q^G \times Q^S$ and terminating at (q_j^G, q_k^S). If $q_j^G = \hat{\delta}^G(q_1, s)$, it follows from Definition 9 that $\mu(\{s\}) = \tilde{\pi}^G[s, q_1^G]\chi^G(q_j^G)$ for the unsupervised (i.e., uncontrolled) plant. Following equations (18) and (19), the measure of the supervised (i.e., controlled) plant becomes

$$\begin{aligned} \mu_1(\{s\}) &= \tilde{\pi}\big[s, \big(q_1^G, q_1^S\big)\big]\,\chi\Big(\Big(q_j^G, q_k^S\Big)\Big) \\ &= \tilde{\pi}^G\big[s, q_1^G\big]\,\chi^G\Big(q_j^G\Big)\,\mathcal{J}\big(q_k^S\big). \end{aligned} \tag{21}$$

In other words, if no event in the string s is disabled by the supervisor, then $\mu_1(\{s\})$ in the supervised plant automaton $\mathbb{G}$ remains the same as in the unsupervised plant automaton G; otherwise, $\mu_1(\{s\}) = 0$. Thus, Definition 11 guarantees that the same strings in different supervised sublanguages of the unsupervised plant language $L(G_i)$ are assigned the same measure. Hence, the performance of different supervisors can be compared with a common quantitative tool.

Finally to conclude this section, it should be noted that while the domain (i.e., $2^{L(G_i)}$) of the language measure μ^i is partially ordered, its range which is a subset of $\mathbf{R}$ becomes totally ordered. The set $L(G_i)$ with the σ-algebra, $2^{L(G_i)}$, forms a measurable space. In principle, any measure μ can be defined on this measurable space to form a measure space (i.e., the triple $\langle L(G_i), 2^{L(G_i)}, \mu^i\rangle$). The choice of the signed language measure, as given by Definitions 9 and 10, has been motivated by the fact that it bears a physical significance and hence is qualified to serve as a performance measure for DES controller synthesis. Moreover, defining the measure in this way also leads to simple computational procedures as discussed in the next section and further elaborated in § 4.

3. Language measure computation

Various methods of obtaining regular expressions for DFSAs are reported in Martin (1997) and Hopcroft *et al.* (2001). While computing the measure of a given DFSA, the same event may have different significance when emanating from different states. This requires assigning (possibly) different costs to the same event defined on different states. Therefore, it is necessary to obtain a regular expression which explicitly yields the state-based event sequences. In order to compute the language measure, it is convenient to transform the procedures of evaluating regular expression from symbolic equations to algebraic ones. The following two methods (Ray and Phoha 2003, Wang and Ray 2004) are presented, in detail, for language measure computation.

3.1. *Method I: closed form solution*

This section presents a closed-form method to compute the language measure via inversion of a square operator.

Definition 12: Let $L_i \equiv L(G_i)$, $i \in \mathcal{I}_Q$, denote the regular expression representing the language of a DFSA $G_i = \langle Q, \Sigma, \delta, q_i, Q_m\rangle$, where q_i is the initial state.

Definition 13: Let σ_j^k denote the set of event(s) $\sigma \in \Sigma$ that is defined on the state q_j and leads to the state $q_k \in Q$, where $j, k \in \mathcal{I}_Q$, i.e., $\delta(q_j, \sigma) = q_k$, $\forall \sigma \in \sigma_j^k \subseteq \Sigma$.

Then, given a DFSA $G_i = \langle Q, \Sigma, \delta, q_i, Q_m\rangle$ the procedure to obtain the system equation by a set of regular expressions L_i of the language $L(G_i)$, $i \in \mathcal{I}_Q$, is as follows:

$$\forall q_i \in Q, \quad L_i = \sum_{j \in \mathcal{I}_Q} R_{i,j} + \epsilon, \tag{22}$$

where the operator $\sum$ indicates the sum of regular expressions (equivalently, union of regular languages); and $R_{i,j}$ is defined as follows.

If there exists $\sigma \in \Sigma$ such that $\delta(q_i, \sigma) = q_j \in Q$ for $j \in \{1, \ldots, n\}$, then $R_{i,j} = \sigma_i^j L_j$, otherwise, $R_{i,j} = \emptyset$.

The set of symbolic equations may be written as

$$L_i = \sum_j \sigma_i^j L_j + \epsilon. \tag{23}$$

The above system of symbolic equations can be solved using a result given below, which is illustrated through an example.

Lemma 1: *Let u, v be two known regular expressions and r be an unknown regular expression that satisfies the following algebraic identity*:

$$r = ur + v. \tag{24}$$

Then, the following relations are true.

(1) *$r = u^* v$ is a solution to equation* (24).
(2) *If $\epsilon \notin u$, then $r = u^* v$ is the unique solution to equation* (24).

Proof: The proof of Lemma 1, which is also known as Arden's relation, is given in Yu (1997) and Ray and Phoha (2003). □

Example 1: In this example, shown in figure 1, the alphabet is $\Sigma = \{a, b\}$; the set of states is $Q = \{1, 2, 3\}$; the initial state is 1; and the only marked state is 2. Let the set of linear algebraic equations representing the transitions at each state of the DFSA be as follows:

$$\left.\begin{aligned} L_1 &= a_1^1 L_1 + b_1^2 L_2 + \epsilon \\ L_2 &= a_2^1 L_1 + b_2^3 L_3 + \epsilon \\ L_3 &= a_3^1 L_1 + b_3^2 L_2 + \epsilon \end{aligned}\right\} \tag{25}$$

where the 'forcing' term ϵ is introduced on the right side of each equation. For example, by application of Lemma 1, the regular expression for the language $L(G_1)$ is given as

$$L_1 = (a_1^1)^* b_1^2 (a_2^1 (a_1^1)^* b_1^2 + b_2^3 a_3^1 (a_1^1)^* b_1^2 + b_2^3 b_3^2)^* + \epsilon.$$

Instead of obtaining regular expressions, the language measure can be directly computed by transforming this set of equations into a system of linear equations based on the following result.

Theorem 1: *Following Definition* 10, *the language measure of the symbolic equation* (23) *is given by*

$$\mu_i = \sum_j \pi_{ij} \mu_j + \chi_i. \tag{26}$$

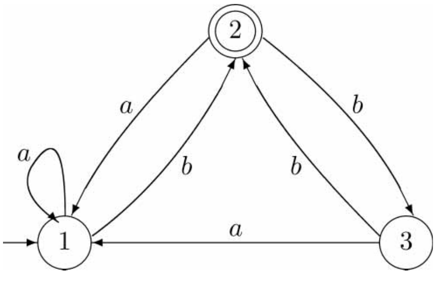


Figure 1. Finite state machine for Example 1.

Proof: Following equation (22) and Definition 5

$$\forall i \in \mathcal{I}_Q, \quad \mu^i(\epsilon) = \chi_i. \tag{27}$$

Therefore, each element of the vector $\mathbf{X} = [\chi_1 \ \chi_2 \ \ldots \ \chi_n]^T$ is the forcing function in equations (23) and (24). Starting from the state q_i, the measure of the language $L_i \equiv L(G_i)$ (see Definition 12)

$$\begin{aligned} \mu_i = \mu^i(L_i) &= \mu^i\left(\sum_j \sigma_i^j L_j + \epsilon\right) \\ &= \mu^i\left(\sum_j \sigma_i^j L_j\right) + \mu^i(\epsilon) \\ &= \sum_j \mu^i(\sigma_i^j L_j) + \nu^i(\varepsilon)\chi_i \\ &= \sum_j \pi(\sigma_i^j)\mu^j(L_j) + \chi_i \\ &= \sum_j \pi_{ij}\mu_j + \chi_i. \end{aligned}$$

The third equality in the above derivation follows from the fact that $\epsilon \cap \sigma_i^j L_j = \emptyset$. It is also true that

$$\forall j \neq k, \qquad \sigma_i^j L_j \bigcap \sigma_i^k L_k = \emptyset \tag{28}$$

since each string in $\sigma_i^j L_j$ starts with an event in σ_i^j while each string in $\sigma_i^k L_k$ starts from an event in σ_i^k and $\sigma_i^j \cap \sigma_i^k = \emptyset \ \forall j \neq k$ because G_i is a DFSA. This justifies the fourth equality. The fifth equality follows from Definition 8 and the fact that $\mu^i(L_{i,j}) = \nu^i(L_{i,j})\chi(q_j)$; therefore, by Definitions 7 and 13, $\mu^i(\sigma_i^j L_j) = \pi[q_i, q_j]$ $\mu^j(L_j) = \pi_{ij}\mu_j$. □

In vector notation, equation (26) in Theorem 1 is expressed as

$$\boldsymbol{\mu} = \mathbf{\Pi}\boldsymbol{\mu} + \mathbf{X}$$

whose solution is given by

$$\boldsymbol{\mu} = (\mathbf{I} - \mathbf{\Pi})^{-1}\mathbf{X} \tag{29}$$

provided that the matrix $\mathbf{I} - \mathbf{\Pi}$ is invertible. The following important result guarantees the existence of $\boldsymbol{\mu}$.

Theorem 2: *Given DFSAs $G_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$, with the state transition cost matrix $\mathbf{\Pi}$, the matrix $(\mathbf{I} - \mathbf{\Pi})$ is an invertible bounded linear operator and $\boldsymbol{\mu} \in \mathbf{R}^n$.*

Proof: It follows from Definitions 6 and 7 that the induced max norm $\|\mathbf{\Pi}\|_\infty \equiv \max_i \sum_j \pi_{ij} = 1 - \theta$ where $\theta \in (0, 1)$. Then $(\mathbf{I} - \mathbf{\Pi})$ is invertible and is a bounded linear operator and $\|(\mathbf{I} - \mathbf{\Pi})^{-1}\|_\infty \leq \theta^{-1}$ (Naylor and

Sell 1982). Then, it follows from equation (29) that $\boldsymbol{\mu} \in \mathbf{R}^n$. □

Corollary 1 (to Theorem 2)**:** *The language measure vector* $\boldsymbol{\mu}$ *is bounded as* $\|\boldsymbol{\mu}\|_\infty \leq \theta^{-1}$ *where* $\theta \equiv (1 - \|\boldsymbol{\Pi}\|_\infty)$.

Proof: The proof follows by applying the norm inequality property and Theorem 2 to equation (29) and the fact that the max norm $\|\mathrm{X}\|_\infty \leq 1$ by Definition 5. □

Alternatively, sufficient conditions for convergence of μ can be obtained based on the properties of nonnegative matrices that are given in Appendix II. Therefore, Definitions 6 and 7 provide a sufficient condition for the language measure μ of the DFSA G_i to be finite. A closed-form algorithm to compute a language measure based on the above procedure is presented below.

Algorithm 1: Closed-form computation of the language measure

(1) For a given $G_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$, specify the characteristic vector $\mathbf{X}$ (see Definition 5) and determine the event cost matrix $\widetilde{\Pi}$ (see Definition 6) via experimentation or simulation, as described later in § 5).
(2) Generate the $\boldsymbol{\Pi}$-matrix (Definition 7).
(3) Compute the language measure vector $\boldsymbol{\mu} \leftarrow (\mathbf{I} - \boldsymbol{\Pi})^{-1}\mathbf{X}$ using Gaussian elimination.
(4) Obtain μ_i, the ith element of $\boldsymbol{\mu}$-vector, which is the measure of the generated language of the DFSA G_i.

The jth element of the ith row of the $(\mathbf{I} - \boldsymbol{\Pi})^{-1}$ matrix, denoted as ν_i^j, is the language measure of the DFSA with the same state transition function δ as G_i and having the following properties: (i) the initial state is q_i; (ii) q_j is the only marked state; and (iii) the χ-value of q_j is equal to 1. Thus, $\mu_i \equiv \mu(L(G_i))$ is given by $\mu_i = \sum_j \nu_i^j \chi_j$. Numerical evaluation of the language measure of the automaton G_i requires Gaussian elimination of the single variable μ_i involving the real invertible matrix $(\mathbf{I} - \boldsymbol{\Pi})$. Therefore, the computational complexity of the language measure algorithm is polynomial in the number of states.

3.2. *Method II: recursive solution*

This section presents a second method to compute the language measure using a recursive procedure based on Kleene's theorem (Martin 1997) which states that the marked (i.e., accepted) language of a DFSA is regular. It also yields an algorithm to recursively construct the regular expression of its language instead of the closed form solution in Method I.

Definition 14: Given q_i, $q_k \in Q$, a non-empty string p of events (i.e., $p \neq \epsilon$) starting from q_i and terminating at q_k is called a path. A path p from q_i to q_k is said to pass through q_j if there exists $s \neq \epsilon$ and $t \neq \epsilon$ such that $p = st$; $\hat{\delta}(q_i, s) = q_j$ and $\hat{\delta}(q_j, t) = q_k$.

Definition 15: A path language p_{ik}^j is defined to be the set of all paths from q_i to q_k, which do not pass through any state q_r for $r > j$, and $\epsilon \notin p_{ik}^j$. The path language p_{ik} is defined to be the set of all paths from q_i to q_k. Thus, the language $L_{i,k}$ is obtained in terms of the path language p_{ik} as

$$L_{i,k} = \begin{cases} p_{ii} \bigcup \{\epsilon\}, & \text{if } k = i \\ p_{ik}, & \text{if } k \neq i \end{cases}$$
$$\Rightarrow \nu(L_{i,k}) = \begin{cases} \nu(p_{ii}) + 1, & \text{if } k = i \\ \nu(p_{ik}), & \text{if } k \neq i. \end{cases}$$

Every path language p_{ik}^j is a regular language and is a subset of $L(G_i)$. As shown in Ray and Phoha (2003), following recursive relation holds for $0 \leq j \leq n-1$, where $\|\mathrm{Q}\| = n$.

Theorem 3: Given a DFSA $G_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$, the following recursive relation holds for $1 \leq j \leq n-1$

$$\left.\begin{aligned} p_{lk}^0 &= \{\sigma \in \Sigma \colon \delta(q_l \sigma) = q_k\} \\ p_{lk}^{j+1} &= p_{lk}^j \bigcup p_{l,j+1}^j \left(p_{j+1,j+1}^j\right)^* p_{j+1,k}^j. \end{aligned}\right\} \tag{30}$$

Proof: Since the states are numbered form 1 to n in increasing order, $p_{lk}^0 = \{\sigma \in \Sigma \colon \delta(q_l, \sigma) = q_k\}$ follows directly form the state transition map $\delta \colon Q \times \Sigma \to Q$ and Definition 15.

Given $p_{lk}^j \subseteq p_{lk}^{j+1}$, let us consider the set $p_{lk}^{j+1} - p_{lk}^j$ in which each string passes through q_{j+1} in the path from q_l to q_k and no string must pass through q_m for $m > (j+1)$. Then, it follows that

$$p_{lk}^{j+1} - p_{lk}^j = p_{l,j+1}^j p_{j+1,k}^{j+1}$$

where $p_{j+1,k}^{j+1}$ can be expanded as

$$p_{j+1,k}^{j+1} = \left(p_{j+1,j+1}^j p_{j+1,k}^{j+1}\right) \bigcup p_{j+1,k}^j$$

that has a unique solution following Theorem 1 because $\epsilon \notin p_{j+1,j+1}^j$ based on Definition 15. Therefore,

$$p_{lk}^{j+1} = p_{lk}^j \bigcup p_{l,j+1}^j \left(p_{j+1,j+1}^j\right)^* p_{j+1,k}^j. \qquad \square$$

Based on the three lemmas proved below, the above relations can be transformed into an algebraic equation conceptually similar to Theorem 1 in Method I. Along with the procedure to compute the language measure it is established that, $\forall i \in \mathcal{I}_Q$, $\sum_{j=1}^n \pi_{ij} < 1$ is a sufficient condition for finiteness of μ.

Lemma 2: $\nu((p^0_{kk})^*(\cup_{j\neq k} p^0_{kj})) \in [0,1)$.

Proof: Following Definitions 6 and 8, $\nu(p^0_{kk}) \in [0,1)$. Therefore, by convergence of geometric series,

$$\nu\left((p^0_{kk})^*\left(\bigcup_{j\neq k} p^0_{kj}\right)\right) = \frac{\sum_{j\neq k}\nu\left(p^0_{kj}\right)}{1-\nu\left(p^0_{kk}\right)} \in [0,1)$$

because $\sum_j \nu(p^0_{kj}) \Rightarrow \sum_{j\neq k}\nu(p^0_{kj}) - \nu(p^0_{kk})$. □

Lemma 3: $\nu(p^j_{j+1,j+1}) \in [0,1)$.

Proof: The path $p^j_{j+1,j+1}$ may contain at most j loops, one around each of the states $q_1, q_2, \ldots, q_j$. If the path $p^j_{j+1,j+1}$ does not contain any loop, then $\nu(p^j_{j+1,j+1}) \in [0,1)$ because $\forall s \in p^j_{j+1,j+1}$, $\nu(s) < 1$ and each of s originates at state $j+1$.

Next let us suppose that there is a loop around q_ℓ and that does not contain any other loop; this loop must be followed by one or more events σ_k generated at q_ℓ and leading to some other states q_m where $m \in \{1,\ldots,j+1\}$ and $m \neq \ell$. By Lemma 2, $\nu(p^j_{j+1,\,j+1}) \in [0,1)$. Proof follows by starting from the innermost loop and ending with all loops at q_j □

Lemma 4:

$$\nu\left((p^j_{j+1,j+1})^*\right) = \frac{1}{1-\nu\left(p^j_{j+1,j+1}\right)} \in [1,\infty). \qquad (31)$$

Proof: Since $\nu(p^j_{j+1,j+1}) \in [0,1)$ from Lemma 3.3, it follows that

$$\nu\left((p^j_{j+1,j+1})^*\right) = \frac{1}{1-\nu\left(p^j_{j+1,j+1}\right)} \in [1,\infty). \quad \square$$

Finally, the main result of this section is stated as the following theorem.

Theorem 4: *Given a DFSA $G_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m\rangle$ the following recursive result holds for $0 \le j \le n-1$, where $\|\mathrm{Q}\| = n$.*

$$\nu\left(p^{j+1}_{lk}\right) = \nu\left(p^j_{lk}\right) + \frac{\nu\left(p^j_{l,j+1}\right)\nu\left(p^j_{j+1,k}\right)}{1-\nu\left(p^j_{j+1,j+1}\right)}. \qquad (32)$$

Proof:

$$\begin{aligned}\nu\left(p^{j+1}_{lk}\right) &= \nu\left(p^j_{lk} \bigcup p^j_{l,j+1}\left(p^j_{j+1,j+1}\right)^* p^j_{j+1,k}\right)\\ &= \nu\left(p^j_{lk}\right) + \nu\left(p^j_{l,j+1}\left(p^j_{j+1,j+1}\right)^* p^j_{j+1,k}\right)\\ &= \nu\left(p^j_{lk}\right) + \nu\left(p^j_{l,j+1}\right)\nu\left(\left(p^j_{j+1,j+1}\right)^*\right)\nu\left(p^j_{j+1,k}\right)\\ &= \nu\left(p^j_{lk}\right) + \frac{\nu\left(p^j_{l,j+1}\right)\nu\left(p^j_{j+1,k}\right)}{1-\nu\left(p^j_{j+1,j+1}\right)}.\end{aligned}$$

The second step in the above derivation follows from fact that $p^j_{lk} \cap p^j_{l,j+1}(p^j_{j+1,j+1})^* p^j_{j+1,k} = \emptyset$. The third step follows from Definition 8 and the last step is a consequence of Lemma 4. □

Based on the above result, a recursive algorithm to compute a language measure is presented below.

Algorithm 2: Recursive computation of the language measure

```
(1) For a given $G_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m\rangle$, specify the characteristic vector **X** (see Definition 5) and determine the event cost matrix $\tilde{\mathbf{\Pi}}$ (see Definition 6) via experimentation or simulation, as described in § 5).
(2) Compute the **Π**-matrix (Definition 7).
(3) $\nu(p^0_{lk}) \longleftarrow \pi_{lk}$ for $1 \le l, k \le n$
(4) for $j = 0$ to $n-1$
      for $l = 1$ to $n$
        for $k = 1$ to $n$
          $\nu(p^{j+1}_{lk}) = \nu(p^j_{lk}) + \frac{\nu(p^j_{l,j+1})\nu(p^j_{j+1,k})}{1-\nu(p^j_{j+1,j+1})}$
        end
      end
    end
(5) Calculate $\nu(L_{i,k})$ from $\nu(p_{ik})$ using Definition 15.
(6) $\mu_i \longleftarrow \sum_{q_j \in Q_m} \nu(L_{i,J})\chi_j$ is a measure of the language $L_i$ of the DFSA $G_i$.
```

Since there are only three *for* loops, the computational complexity of the above algorithm is polynomial in the number of DFSA states, same as that of Algorithm 1 in Method I.

4. Event cost: a probabilistic interpretation

The signed real measure (see Definition 10) of a regular language is based on the assignment of the characteristic vector **X** (see Definition 5) and the event cost matrix $\tilde{\mathbf{\Pi}}$ (Definition 6). The characteristic vector is chosen by the designer based on his/her perception of the individual state's impact on the system performance. On the other hand, the event cost is an intrinsic property of the plant. The event cost $\tilde{\pi}_{jk}$ is conceptually similar to the state-based conditional probability of Markov Chains, except for the fact that it is not allowed to satisfy the equality condition $\sum_k \tilde{\pi}_{jk} = 1$. (Note that $\sum_k \tilde{\pi}_{jk} < 1$ is a requirement for convergence of the language measure.) The rationale for this strict inequality is explained below.

Since the plant model is an inexact representation of the physical plant, there exist unmodelled dynamics to account for. This can manifest itself either as unmodelled events that may occur at each state or as unaccounted states in the model. Let Σ^u_j denote the set of all unmodelled events at state q_j of the DFSA $G_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$. Creating a new unmarked absorbing state q_{n+1}, called the dump state (Ramadge and Wonham 1987), and extending the transition function δ to $\delta_{\text{ext}}: (Q \cup \{q_{n+1}\}) \times (\Sigma \cup_j \Sigma_j^u) \to (Q \cup \{q_{n+1}\})$, it follows that

$$\delta_{\text{ext}}(q_j, \sigma) = \begin{cases} \delta(q_j, \sigma), & \text{if } q_j \in Q \text{ and } \sigma \in \Sigma \\ q_{n+1}, & \text{if } q_j \in Q \text{ and } \sigma \in \Sigma_j^u \\ q_{n+1}, & \text{if } j = n+1 \text{ and } \sigma \in \Sigma \bigcup \Sigma_j^u. \end{cases} \tag{33}$$

Therefore the residue $\theta_j = 1 - \sum_k \tilde{\pi}_{jk}$ denotes the probability of the set of unmodelled events Σ_j^u conditioned on the state j. The $\mathbf{\Pi}$ matrix can be similarly augmented to obtain a stochastic matrix $\mathbf{\Pi}_{\text{aug}}$ as follows:

$$\mathbf{\Pi}_{\text{aug}} = \begin{bmatrix} \pi_{11} & \pi_{12} & \dots & \pi_{1n} & \theta_1 \\ \pi_{21} & \pi_{22} & \dots & \pi_{2n} & \theta_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \pi_{n1} & \pi_{n2} & \dots & \pi_{nn} & \theta_n \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}. \tag{34}$$

Since the dump state q_{n+1} is not marked, its characteristic value $\chi_{n+1} \equiv \chi(q_{n+1}) = 0$. The characteristic vector then augments to

$$\mathbf{X}_{\text{aug}} = [\mathbf{X}^T \quad 0]^T$$

and, with these extensions, the language measure vector $\boldsymbol{\mu}_{\text{aug}} \equiv [\mu_1 \ \mu_2 \ \cdots \ \mu_n \ \mu_{n+1}]^T = [\boldsymbol{\mu}^T \ \ \mu_{n+1}]^T$ of the augmented DFSA $G_{\text{aug}} \equiv \langle Q \cup \{q_{n+1}\}, \Sigma \cup_j \Sigma_j^u, \delta_{\text{ext}}, q_i, Q_m \rangle$ can be expressed as

$$\boldsymbol{\mu}_{\text{aug}} \equiv \begin{pmatrix} \boldsymbol{\mu} \\ \mu_{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{\Pi}\boldsymbol{\mu} + \mu_{n+1}[\theta_1 \quad \cdots \quad \theta_n]^T \\ \mu_{n+1} \end{pmatrix} + \begin{pmatrix} \mathbf{X} \\ 0 \end{pmatrix}. \tag{35}$$

Since $\chi(q_{n+1}) = 0$ and all transitions from the absorbing state q_{n+1} lead to itself, i.e., $\mu_{n+1} = \mu(L_m(G_{n+1})) = 0$, equation (35) reduces to that for the original plant G_i. Thus, the event cost can now be interpreted as conditional probability, where the residue $\theta_j = 1 - \sum_k \tilde{\pi}_{jk} > 0$ accounts for the probability of all unmodelled events emanating from the state q_j. With this interpretation of event cost, $\tilde{\pi}[s, q_i]$ (see Definition 6) denotes the probability of occurrence of the event string s in the plant model G_i starting at state q_i and terminating at state $\hat{\delta}(s, q_i)$. Hence, $\nu^i(L_{i,j})$ (see Definition 8), which is a non-negative real number, is directly related to the sum of probabilities that state q_i would be reached via alternative paths as the plant operates. (Note that $\nu^i(L_{i,j}) > 1$ is possible if $L_{i,j}$ contains multiple strings.) The language measure $\mu_i \equiv \mu^i(L(G_i)) = \sum_{j \in \mathcal{I}_Q} \mu^i(L_{i,j}) = \sum_{j \in \mathcal{I}_Q} \nu^i(L_{i,j})\chi_j$ is then directly related (but not necessarily equal) to the expected value of the characteristic function.

The choice of the characteristic function (see Definition 5) is based on the importance assigned to the individual marked states of the DFSA. Therefore, in the setting of the language measure, a supervisor's performance is superior if the supervised plant is more likely to terminate at a good marked state and/or less likely to terminate at a bad marked state.

5. Estimation of language measure parameters

This section presents a recursive algorithm for identification of the language measure parameters (Wang *et al.* 2005) (i.e., elements of the event cost matrix $\widetilde{\mathbf{\Pi}}$) (see Definition 6) which, in turn, allows computation of the state transition cost matrix $\mathbf{\Pi}$ (see Definition 7) and the language measure $\boldsymbol{\mu}$-vector (see Definition 10). It is assumed that the underlying physical process evolves at two different time scales. In the fast-time scale, i.e., over a short time period, the system is assumed to be an ergodic, discrete Markov process. In the slowly-varying time scale, i.e., over a long period, the system (possibly) behaves as a non-stationary stochastic process. For such a slowly-varying non-stationary process, it might be necessary to redesign the supervisory control policy in real time. In that case, the $\widetilde{\mathbf{\Pi}}$-matrix parameters should be updated at selected slow-time epochs.

5.1. *A recursive parameter estimation scheme*

Let p_{ij} be the transition probability of the event σ_j at the state q_i, i.e.,

$$p_{ij} = \begin{cases} P[\sigma_j | q_i], & \text{if } \exists q \in Q, \text{ s.t. } q = \delta(q_i, \sigma_j) \\ 0, & \text{otherwise} \end{cases} \tag{36}$$

and its estimate be denoted by the parameter $\hat{p}_{ij}$ that is to be identified from the ensemble of simulation and/or experimental data.

Let a strictly increasing sequence of time epochs of consecutive event occurrence be denoted as

$$\mathcal{T} \equiv \{t_k: k \in N_0\}, \tag{37}$$

where $\mathbf{N}_0$ is the set of non-negative integers. Let the indicator $\psi: \mathbf{N}_0 \times \mathcal{I}_Q \times \mathcal{I}_\Sigma \to \{0, 1\}$ represent the incident of occurrence of an event. For example, if the DFSA was in state q_i at time epoch t_{k-1}, then

$$\psi_{ij}(k) = \begin{cases} 1, & \text{if } \sigma_j \text{ occurs at the time epoch } t_k \in \mathcal{T} \\ 0, & \text{otherwise.} \end{cases} \tag{38}$$

Consequently, the number of occurrences of any event in the alphabet Σ is represented by $\Psi: \mathbf{N}_0 \times \mathcal{I}_Q \to \{0,1\}$. For example, if the DFSA was in state q_i at the time epoch t_{k-1}, then

$$\mathbf{\Psi}_i(k) = \sum_{j \in \mathcal{I}_\Sigma} \psi_{ij}(k). \tag{39}$$

Let n: $\mathbf{N}_0 \times \mathcal{I}_Q \times \mathcal{I}_\Sigma \to \mathbf{N}_0$ represent the cumulative number of occurrences of an event at a state up to a given time epoch. That is, $n_{ij}(k)$ denotes the number of occurrences of the event σ_j at the state q_i up to the time epoch $t_k \in \mathcal{T}$. Similarly, let N: $\mathbf{N}_0 \times \mathcal{I}_Q \to \mathbf{N}_0$ represent the cumulative number of occurrences of any event in the alphabet Σ at a state up to a given time epoch. Consequently,

$$N_i(k) = \sum_{j \in \mathcal{I}_\Sigma} n_{ij}(k). \tag{40}$$

A frequency estimator, $\hat{p}_{ij}(k)$, for probability $p_{ij}(k)$ of the event σ_j occurring at the state q_i at the time epoch t_k, is obtained as

$$\left.\begin{aligned} \hat{p}_{ij}(k) &= \frac{n_{ij}(k)}{N_i(k)} \\ \lim_{k\to\infty} \hat{p}_{ij}(k) &= p_{ij}. \end{aligned}\right\} \tag{41}$$

Convergence of the above limit is justified because the occurrence of an event at a given state of a stationary Markov chain can be treated as an independent and identically distributed random variable.

A recursive algorithm of learning p_{ij} is formulated as a stochastic approximation scheme, starting at the time epoch t_0 with the initial conditions: $\hat{p}_{ij}(0) = 0$ and $n_{ij}(0) = 0$ for all $i \in \mathcal{I}_Q, j \in \mathcal{I}_\Sigma$; and $\mathbf{\Psi}_i(0) = 0$ for all $i \in \mathcal{I}_Q$. Starting at $k = 0$, the recursive algorithm runs for $\{t_k: k \geq 1\}$. For example, upon occurrence of an event σ_j at a state q_i, the algorithm is recursively incremented as

$$\left.\begin{aligned} n_{ij}(k) &= n_{ij}(k-1) + \psi_{ij}(k) \\ N_i(k) &= N_i(k-1) + \Psi_i(k). \end{aligned}\right\} \tag{42}$$

Next it is demonstrated how the estimates of the language parameters (i.e., the elements of event cost matrix $\tilde{\Pi}$) are determined from the probability estimates. As stated earlier in §4 the set of unmodelled events at state q_i, denoted by $\Sigma_i^u \ \forall i \in \mathcal{I}_Q$, accounts for the row-sum inequality: $\sum_j \tilde{\pi}_{ij} <$ (see Definition 6). Then, $\mathrm{P}[\Sigma_i^u] = \theta_i \in (0,1]$ and $\sum_i \tilde{\pi}_{ij} = 1 - \theta_i$. An estimate of the (i,j)th element of the event cost matrix $\tilde{\mathbf{\Pi}}$-matrix, denoted by $\hat{\tilde{\pi}}_{ij}$, is approximated as

$$\hat{\tilde{\pi}}_{ij}(k) = \hat{p}_{ij}(k)(1-\theta_i) \quad \forall j \in \mathcal{I}_\Sigma. \tag{43}$$

Additional experiments on a more detailed automaton model would be necessary to identify the parameters $\theta_i \ \forall i \in \mathcal{I}_Q$. If $\theta_i \ll 1$, the problem of conducting additional experimentation can be circumvented by the following approximation.

A single parameter $\theta \approx \theta_i \ \forall i \in \mathcal{I}_Q$, $i \in \mathcal{I}_Q$, such that $0 < \theta \ll 1$, could be selected for convenience of implementation. From the numerical perspective, this option is meaningful because it sets an upper bound on the language measure based on the fact that the max norm $\|\mu\|_\infty \leq \theta^{-1}$. Note that each row sum in the $\tilde{\mathbf{\Pi}}$-matrix being strictly less than 1, i.e., $\sum_j \tilde{\pi}_{ij} < 1$, is a sufficient condition for finiteness of the language measure (see Appendix II).

Theoretically, $\tilde{\pi}_{ij}$ is the asymptotic value of the estimated probabilities $\hat{\tilde{\pi}}_{ij}(k)$ as if the event σ_j occurs infinitely many times at the state q_i. However, while dealing with finite amount of data, the objective is to obtain a good estimate $\hat{p}_{ij}$ of p_{ij} from independent Bernoulli trials of generating events. Critical issues in this situation are: (i) how much data are needed; and (ii) when to stop if adequate data are available. The next section 5-B addresses these issues.

5.2. *Stopping rules for recursive learning*

A stopping rule is necessary to find a lower bound on the number of experiments to be conducted for identification of the $\tilde{\mathbf{\Pi}}$-matrix parameters. This section presents two stopping rules that are discussed below.

The first stopping rule is based on an inference approximation having a specified absolute error bound ε with a probability λ. The objective is to achieve a trade-off between the number of experimental observations and the estimation accuracy.

A bound on the required number of samples is estimated using the Gaussian structure of the binomial distribution that is an approximation of the sum of a large number of independent and identically distributed (i.i.d.) Bernoulli trials of $\hat{\tilde{\pi}}_{ij}(t)$. The central limit theorem yields $\hat{\tilde{\pi}}_{ij} \sim \mathcal{N}(\tilde{\pi}_{ij}, \tilde{\pi}_{ij}(1-\tilde{\pi}_{ij})/N)$, where $\mathcal{N}$ indicates normal (or Gaussian) distribution with $E[\hat{\tilde{\pi}}_{ij}] \approx \tilde{\pi}_{ij}$ and $\mathrm{Var}[\hat{\tilde{\pi}}_{ij}] \equiv \sigma^2 \approx \tilde{\pi}_{ij}(1-\tilde{\pi}_{ij})/N$, provided that the number of samples N is sufficiently large. Let $\Delta = \hat{\tilde{\pi}}_{ij} - \tilde{\pi}_{ij}$, then $\Delta/\sigma \sim \mathcal{N}(0,1)$. Given $0 < \varepsilon \ll 1$ and $0 < \lambda \ll 1$, the problem is to find a bound N_b on the number N of experiments such that $P\{|\Delta| \geq \varepsilon\} \leq \lambda$. Equivalently,

$$P\left\{\frac{|\Delta|}{\sigma} \geq \frac{\varepsilon}{\sigma}\right\} \leq \lambda \tag{44}$$

that yields a bound N_b on N as

$$N_b \geq \left(\frac{\xi^{-1}(\lambda)}{\varepsilon}\right)^2 \tilde{\pi}_{ij}(1-\tilde{\pi}_{ij}), \tag{45}$$

where $\xi(x) \equiv 1 - \sqrt{2/\pi}\int_0^x e^{-(t^2/2)}\,dt$. Since the parameter $\tilde{\pi}_{ij}$ is unknown, one may use the fact that $\tilde{\pi}_{ij}(1-\tilde{\pi}_{ij}) \leq 0.25$ for every $\tilde{\pi}_{ij} \in [0,1]$ to (conservatively) obtain a bound on N only in terms of the specified parameters ε and λ as

$$N_b \geq \left(\frac{\xi^{-1}(\lambda)}{2\varepsilon}\right)^2. \tag{46}$$

The above estimate of the bound on the required number of samples is less conservative than that obtained from the Chernoff bound and is significantly less conservative than that obtained from Chebyshev bound which does not require the assumption of any specific distribution of Δ except for finiteness of the rth ($r = 2$) moment.

The second stopping rule, which is an alternative to the first stopping rule, is based on the properties of irreducible stochastic matrices. Following equation (41) and the state transition function δ of the DFSA, the state transition matrix is constructed at the kth iteration as $\mathbf{P}(k)$ that is an $n \times n$ irreducible stochastic matrix under stationary conditions. Similarly, the state probability vector $\mathbf{p}(k) \equiv [p_1(k)\ p_2(k)\ \cdots\ p_n(k)]$ is obtained by following equation (41)

$$p_i(k) = \frac{N_i(k)}{\sum_{j \in \mathcal{I}_Q} N_j(k)}. \tag{47}$$

The stopping rule makes use of the Perron–Frobenius Theorem to establish a relation between the vector $\mathbf{p}(k)$ and the irreducible stochastic matrix $\mathbf{P}(k)$.

Theorem 5: *Perron–Frobenius Theorem (Senata 1973, Plemmons and Berman 1979) Let* $\mathbf{P}(k)$ *be an* $n \times n$ *irreducible matrix, then there exits an eigenvalue r such that*

(1) $r \in \mathbb{R}$ *and* $r > 0$.
(2) *r can be associated strictly positive left and right eigenvectors.*
(3) $r \geq \lambda\ \forall$ *eigenvalue* $\lambda \neq r$.
(4) *The eigenvectors associated with r are unique to constant multiples.*
(5) *If* $0 \leq B \leq P(k)$ *and* β *is an eigenvalue of B, then* $|\beta| \leq |r|$. *Moreover,* $|\beta| = r$ *implies* $B = P(k)$.
(6) *r is a simple root of the characteristic equation of* $P(k)$.

Corollary 1: *Corollary to Perron–Frobenius Theorem*

$$\min_i \sum_{j=1}^{n} P_{ij}(k) \leq r \leq \max_i \sum_{j=1}^{n} P_{ij}(k)$$

with equality on either side implying equality throughout.

Since $\mathbf{P}(k)$ is a stochastic matrix, i.e., $\sum_{j=1}^{n} P_{ij}(k) = 1$, and $\mathbf{P}(k)$ is irreducible, there is a unique eigenvalue $r = 1$ and the corresponding left eigenvector $\mathbf{p}(k)$ (normalized to unity in the sense of absolute sum) representing the state probability vector, provided that the matrix parameters have converged after sufficiently large number of iterations. That is,

$$\|\mathbf{p}(k)(\mathbf{I} - \mathbf{P}(k))\|_\infty \leq \frac{1}{k} \to 0 \quad \text{as } k \to \infty.$$

Equivalently,

$$\|(\mathbf{p}(k) - \mathbf{p}(k+1))\|_\infty \leq \frac{1}{k} \to 0 \quad \text{as } k \to \infty. \tag{48}$$

Taking the expected value of $\|\mathbf{p}(k)\|_\infty$ to be $1/n$, a threshold of η/n is specified, where n is the number of states and $0 < \eta \ll 1$ is a constant. A lower bound on the required number of samples is determined from equation (48) as

$$N_{\text{stop}} \equiv \text{Integer}\left(\frac{n}{\eta}\right) \tag{49}$$

based on the number of states, n, and the specified tolerance η.

6. Usage of the language measure

The two methods of language measure computation, presented in § 3, have the same computational complexity, $\mathcal{O}(n^3)$, where n is the number of states of the DFSA. However, each of these two methods offer distinct relative advantages in specific contexts. For example, while the closed form solution in § 3.1 is more amenable for analysis and synthesis of decision and control algorithms, the recursive solution in § 3.2 might prove very useful for construction of executable codes in real time applications. The following two subsections present usage of the language measure for construction of metric spaces of formal languages and synthesis of optimal discrete-event supervisors.

6.1. *Vector space of formal languages*

The language measure can be used to construct a vector space of sublanguages for a given DFSA $G_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$. The total variation measure $|\mu|$ (Rudin 1987) (see Appendix I) induces a metric on this space, which quantifies the distance function between any two sublanguages of $L(G_i)$.

Proposition 1: *Let* $L(G_i)$ *be the language of a DFSA* $G_i = \langle Q, \Sigma, \delta, q_i, Q_m \rangle$. *Let the binary operation of* *exclusive-OR* $\oplus$: $2^{L(G_i)} \times 2^{L(G_i)} \to 2^{L(G_i)}$ *be defined as*

$$(K_1 \oplus K_2) = \left(K_1 \bigcup K_2\right) - \left(K_1 \bigcap K_2\right) \tag{50}$$

$\forall K_1, K_2 \subseteq L(G_i)$. *Then* $(2^{L(G_i)}, \oplus)$ *is a vector space over Galois field* $GF(2)$.

Proof: It follows from the properties of exclusive-OR that the algebra $(2^{L(G_i)}, \oplus)$ is an Abelian group where $\emptyset$ is the zero element of the group and the unique inverse of every element $K \subseteq 2^{L(G_i)}$ is K itself because $K_1 \oplus K_2 = \emptyset$ if and only if $K_1 = K_2$. The associative and distributive properties of the vector space follows by defining the scalar multiplication of vectors as: $0 \otimes K = \emptyset$ and $1 \otimes K = K$. □

The collection of singleton languages made from each element of $L(G_i)$ forms a basis set of vector space $(2^{L(G_i)}, \oplus)$ over $GF(2)$. It is shown below, how total variation (Rudin 1987) of the signed measure μ can be used to define a metric on above vector space.

Proposition 2: *Total variation measure* $|\mu|$ *on* $2^{L(G_i)}$ *is given by* $|\mu|(L) = \sum_{s\in L} |\mu(\{s\})|$ $\forall L \subseteq L(G_i)$.

Proof: The proof follows from the fact that $\Sigma_k|\mu(L_k)|$ attains its supremum for the finest partition of L which consists of the individual strings in L as elements of the partition. □

Corollary 2 (to Proposition 2)**:** *Let* $L(G_i)$ *be the language of a DFSA* $G_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m\rangle$. *For any* $K \in 2^{L(G_i)}$, $|\mu|(K) \le \theta^{-1}$ *where* $\theta = 1 - \|\boldsymbol{\Pi}\|_\infty$ *and* $\boldsymbol{\Pi}$ *is the state transition cost matrix of the DFSA*.

Proof: The proof follows from Proposition 2 and Corollary 1. □

Definition 16: Let $L(G_i)$ be the language of a DFSA $G_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m\rangle$. The distance function $d: 2^{L(G_i)} \times 2^{L(G_i)} \to [0, \infty)$ is defined in terms of the total variation measure as $\forall K_1, K_2 \subseteq L(G_i)$

$$d(K_1, K_2) = |\mu|\left(\left(K_1 \bigcup K_2\right) - \left(K_1 \bigcap K_2\right)\right) \tag{51}$$

The above distance function $d(\cdot, \cdot)$ quantifies the difference between two supervisors relative to the supervised performance of the DFSA plant.

Proposition 3: *The distance function defined above is a pseudo-metric on the space* $2^{L(G_i)}$

Proof: Since the total variation of a signed real measure is bounded (Rudin 1987), $\forall K_1, K_2 \subseteq L(G_i)$, $d(K_1, K_2) = |\mu|(K_1 \oplus K_2) \in [0, \infty)$; also by Definition 16, $d(K_1, K_2) = d(K_2, K_1)$. The remaining property of the triangular inequality follows from the inequality $|\mu|(K_1 \oplus K_2) \le |\mu|(K_1) + |\mu|(K_2)$ which is based on the fact that $(K_1 \oplus K_2) \subseteq (K_1 \cup K_2)$ and $|\mu|(K_1) \le |\mu|(K_2)$ $\forall K_1 \subset K_2$. □

The pseudo-metric $|\mu|: 2^{L(G_i)} \to [0, \infty)$ can be converted to a metric of the space $(2^{L(G_i)}, \oplus)$ by clustering all languages that have zero total variation measure as the null equivalence class $\mathcal{N} \equiv \{K \in 2^{L(G_i)}: |\mu|(K) = 0\}$. This procedure is conceptually similar to what is done for defining norms in the L_p spaces. In that case, $\mathcal{N}$ contains all sublanguages of $L(G_i)$, which terminate on non-marked states starting from the initial state, i.e., $\mathcal{N} = \{\emptyset \cup (\cup_{q_j \notin Q_m} L_{i,j})$. In the sequel, $|\mu|(\cdot)$ is referred to as a metric of the space $2^{L(G_i)}$. Thus, the metric $|\mu|(\cdot)$ can be generated from $d(\cdot, \cdot)$ as: $|\mu|(K) = d(K, J)$ $\forall K \in 2^{L(G_i)}$ $\forall J \in \mathcal{N}$. Unlike the norms on vector spaces defined over infinite fields, the metric $|\mu|(\cdot)$ for the vector space $(2^{L(G_i)}, \oplus)$ over $GF(2)$ is not a functional. This interpretation of language as a vector and associating a metric to quantify distance between languages, may be useful for analysis and synthesis of discrete-event supervisory (DES) control systems under different settings.

6.2. *Optimal control of regular languages*

The (signed) language measure μ could serve as the performance index for synthesis of an optimal control policy that maximizes the performance of a supervised sublanguage. The salient concept is briefly presented below.

Let $\mathcal{S} \equiv \{S^0, S^1, \ldots, S^N\}$ be a set of supervisory control policies for the unsupervised plant automaton G where S^0 is the null controller (i.e., no event is disabled) implying that $L(S^0/G) = L(G)$. Therefore, the controller cost matrix $\boldsymbol{\Pi}(S^0) = \boldsymbol{\Pi}^0$ that is the $\boldsymbol{\Pi}$-matrix of the unsupervised plant automaton G. For a supervisor $S^k, k \in \{1, 2, \ldots, N\}$, the control policy is required to selectively disable certain controllable events so that the following (elementwise) inequality holds.

$$\boldsymbol{\Pi}^k \equiv \boldsymbol{\Pi}(S^k) \le \boldsymbol{\Pi}^0 \text{ and } L(S^k/G) \subseteq L(G), \quad \forall S^k \in \mathcal{S}.$$

The task is to synthesize an optimal cost matrix $\boldsymbol{\Pi}^* \le \boldsymbol{\Pi}^0$ that maximizes the performance vector $\boldsymbol{\mu}^* \equiv [\mathbf{I} - \boldsymbol{\Pi}^*]^{-1}\mathbf{X}$, i.e., $\boldsymbol{\mu}^* \ge \boldsymbol{\mu}^k \equiv [\mathbf{I} - \boldsymbol{\Pi}^k]^{-1}\mathbf{X}$ $\forall$ $\boldsymbol{\Pi}^k \le \boldsymbol{\Pi}^0$ where the inequalities are implied elementwise. While the details of the underlying theory are available in recent literature (Ray *et al.* 2004), a synthesis procedure for optimal control of regular languages is succinctly presented below.

Let the DFSA model G of the unsupervised plant have the state transition cost matrix: $\boldsymbol{\Pi}^0 \equiv \boldsymbol{\Pi}$ (see Definition 7) and the characteristic vector $\mathbf{X}$ (see Definition 5). Then, the performance vector at the iteration $k = 0$ is given as

$$\boldsymbol{\mu}^0 = [\mu_1^0 \quad \mu_2^0 \quad \ldots \quad \mu_n^0]^T = (I - \boldsymbol{\Pi}^0)^{-1}\mathbf{X}$$

where the jth element μ_j^0 of the vector μ^0 is the performance of the unsupervised plant language, with state q_j as the initial state. Then, $\mu_j^0 < 0$ implies that, if the state q_j is reached, then the plant will yield bad performance thereafter. Intuitively, the control system should attempt to prevent the automaton from reaching q_j by disabling all controllable events that lead to this state. Therefore, the optimal control algorithm starts with disabling all controllable events that lead to every state q_j for which $\mu_j^0 < 0$. This is equivalent to reducing all elements of the corresponding columns of the $\boldsymbol{\Pi}^0$-matrix by disabling those controllable events. In the next iteration, i.e., $k = 1$, the updated cost matrix $\boldsymbol{\Pi}^1$ is obtained as: $\boldsymbol{\Pi}^1 = \boldsymbol{\Pi}^0 - \boldsymbol{\Delta}^0$ where $\boldsymbol{\Delta}^0 \geq 0$ (the inequality being implied elementwise) is composed of event costs corresponding to all controllable events that have been disabled.

It has been shown in Ray *et al.* (2004) that $\boldsymbol{\mu}^{k-1} \leq \boldsymbol{\mu}^k \equiv [I - \boldsymbol{\Pi}^k]^{-1}\,\mathbf{X}$ elementwise for all $k \geq 1$. Although all controllable events leading to every state corresponding to a negative element of $\boldsymbol{\mu}^1$ are disabled, some of the controllable events that were disabled at $k = 0$ may now lead to states corresponding to positive elements of $\boldsymbol{\mu}^1$. Performance could be further enhanced by re-enabling these controllable events. For $k \geq 1$, $\boldsymbol{\Pi}^{k+1} = \boldsymbol{\Pi}^k + \boldsymbol{\Delta}^k$ where $\boldsymbol{\Delta}^k \geq 0$ is composed of the state transition costs of all re-enabled controllable events at k. It is also shown in Ray *et al.* (2004) that the number of iterations to reach optimality does not exceed the number, n, of *DFSA* states. Therefore, the computational complexity of the optimal control algorithm is polynomial n.

In the optimal control algorithm in Ray *et al.* (2004), if $\boldsymbol{\mu}^0 \geq 0$, i.e., there is no state q_j such that $\boldsymbol{\mu}_j^0 < 0$, then the plant performance cannot be improved by event disabling and the null controller S^0 (i.e., no disabled event) is the optimal controller for the given plant. Therefore, the cases are considered where μ_j^0 for some state q_j.

Starting with $k = 0$ and $\boldsymbol{\Pi}^0 \equiv \boldsymbol{\Pi}^{\text{plant}}$, the control policy is constructed by the following two-step procedure:

Step 1: For every state q_j for which $\mu_j^0 < 0$, disable controllable events leading to q_j. Now, $\boldsymbol{\Pi}^1 = \boldsymbol{\Pi}^0 - \boldsymbol{\Delta}^0$, where $\boldsymbol{\Delta}^0 \geq 0$ is composed of event costs corresponding to all controllable events, leading to q_j for which $\boldsymbol{\mu}_j^0$, which have been disabled at $k = 0$.

Step 2: For $k \geq 1$, if $\mu_j^k \geq 0$, re-enable all controllable events leading to q_j, which were disabled in Step 1. The cost matrix is updated as: $\boldsymbol{\Pi}^{k+1} = \boldsymbol{\Pi}^k + \boldsymbol{\Delta}^k$ for $k \geq 1$, where $\boldsymbol{\Delta}^k \geq 0$ is composed of event costs corresponding to all currently re-enabled controllable events. The iteration is terminated if no controllable event leading to q_j remains disabled for which $\mu_j^k > 0$. At this stage, the optimal performance $\boldsymbol{\mu}^* \equiv [I - \boldsymbol{\Pi}^*]^{-1}\mathbf{X}$.

7. An application example

Ray and Phoha (2003) and Ray *et al.* (2004) have adopted the closed form method of language measure (see § 3.1) as a performance index for optimal supervisory control of a twin-engine unmanned aircraft that is used for surveillance and data collection. The language measure was computed and verified based on both closed form and recursive techniques given in § 3; the results were identical as expected. Engine health and operating conditions, which are monitored in real time based on observed data, are classified into three mutually exclusive and exhaustive categories:

- good;
- unhealthy (but operable);
- inoperable.

In the event of any observed abnormality, the supervisor may decide to continue or abort the mission. The finite state automaton model of the plant in figure 2 has 13 states (excluding the dump state), of which three are marked states, and nine events, of which four are controllable and the remaining five are uncontrollable. All events are assumed to be observable. The states and events of the plant model are listed in table 1 and table 2, respectively. The state transition function δ and the state-based event cost $\tilde{\pi}_{ij}$ (see Definition 6) are entered simultaneously in table 3. The values of $\tilde{\pi}_{ij}$ were selected by extensive experiments on engine simulation models and were also based on experience of gas

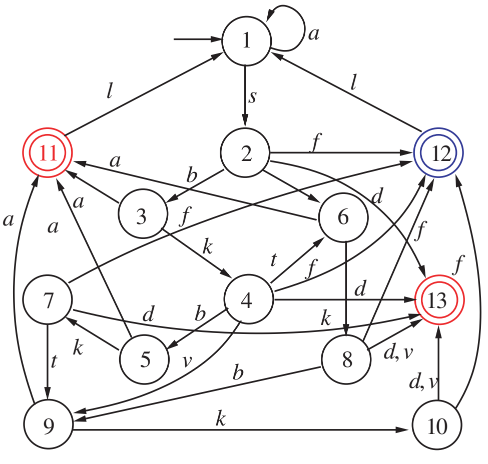


Figure 2. Unsupervised plant, i.e., no disabling of controllable events.

turbine engine operation and maintenance. The dump state and any transitions to the dumped state are not shown in table 3. The elements of the characteristic vector (see Definition 5) were chosen as signed real weights based on the perception of each marked state's role on the engine performance.

Table 1. Plant automation states.

State	Description
1	Safe in base
2	Mission executing — two good engines
3	One engine unhealthy during mission execution
4	Mission executing — one good and one unhealthy engine
5	Both engines unhealthy during mission execution
6	One engine good and one engine inoperable
7	Mission execution with two unhealthy engines
8	Mission execution with only one good engine
9	One engine unhealthy and one engine inoperable
10	Mission execution with only one unhealthy engine
11	Mission aborted/not completed (Bad Marked State)
12	Mission successful (Good Marked State)
13	Aircraft destroyed (Bad Marked State)

Table 2. Plant event alphabet.

Event	Event description	Controllable event
s	Start and take-off	√
b	A good engine becoming unhealthy	
t	An unhealthy engine becoming inoperable	
v	A good engine becoming inoperable	
k	Keep engine(s) running	√
a	Mission abortion	√
f	Mission completion	
d	Destroyed aircraft	
l	Landing	√

The characteristic (χ) values of the 13 states are assigned as: $\mathbf{X} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ {-0.05}\ {+0.25}\ {-1.0}]^T$. These parameters are selected by the designer based on his/her perception of each marked state's role in the system performance. As the states 1 to 10 in in table 1 are not marked, the first 10 elements of the characteristic vector **X** are zeros. The implication is that event strings terminating at states 1 to 10 have no bearing on the system performance and hence have zero measure. The state 12 is a good marked state having a positive χ value and the bad marked states 11 and 13 have negative χ values. Therefore, event strings terminating at state 12 have positive measure and those terminating at states 11 and 13 have negative measure.

Four supervisory controllers were designed independently using a graphical interactive package (Wang *et al.* 2003) based on the following specifications.

(1) Specification #1: At least one of the two engines must be in good condition for mission continuation.
(2) Specification #2: None of the two engines must be in inoperable condition for mission continuation.
(3) Specification #3: Both engines must be in good condition for mission continuation.
(4) Optimal Control: The control policy is optimized following the two-step procedure in § 6.2.

The supervised plant automata under specifications #1, #2, #3 and optimal control are displayed in figures 3, 4, 5 and 6, respectively, where dashed lines indicate disabled controllable events. Notice that none of the controllable events are disabled in the unsupervised plant (see figure 2) and the four supervisors disable different sets of controllable events, as seen in figures 3–6.

Table 3. State transition and event cost matrix.

	s	*b*	*t*	*v*	*k*	*a*	*f*	*d*	*l*
1	0.50(2)					0.02(1)			
2		0.05(3)		0.01(6)			0.80(12)	0.10(13)	
3					0.45(4)	0.45(11)			
4		0.12(5)	0.16(6)	0.10(9)			0.50(12)	0.12(13)	
5					0.45(7)	0.45(11)			
6					0.45(8)	0.45(11)			
7			0.25(9)				0.50(12)	0.20(13)	
8		0.20(9)		0.01(13)			0.3(12)	0.4(13)	
9					0.45(10)	0.45(11)			
10			0.35(13)				0.20(12)	0.40(13)	
11									0.95(1)
12									0.95(1)
13									

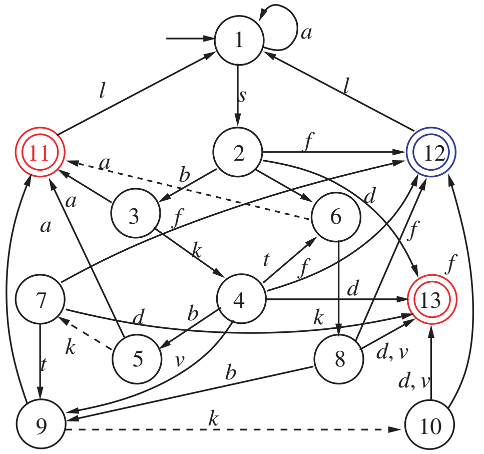


Figure 3. Supervised plant under specification #1.

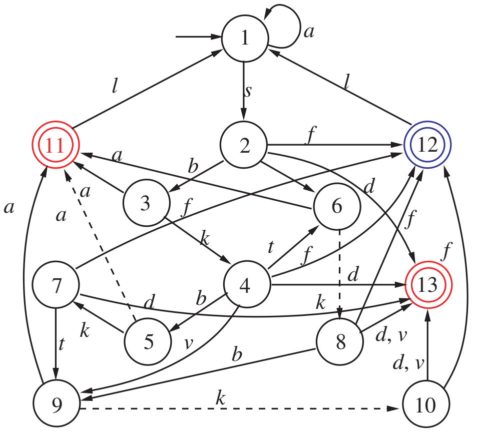


Figure 4. Supervised plant under specification #2.

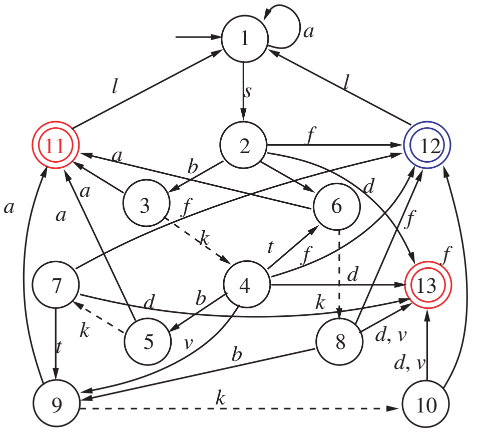


Figure 5. Supervised plant under specification #3.

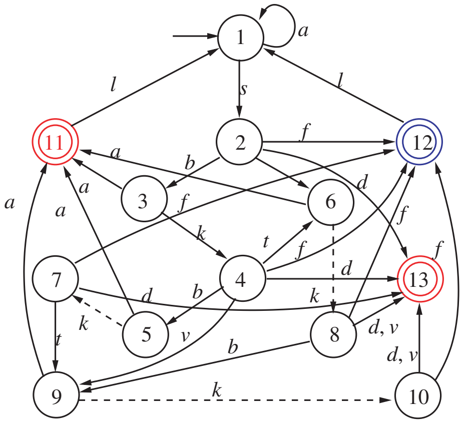


Figure 6. Optimally supervised plant, i.e., with best performance.

The performance measure μ_1 (i.e., with the initial state 1) of the unsupervised (i.e., no disabling of control events) plant is 0.0823 and for three supervised plants under specifications #1, #2, #3 and the optimally supervised plant are evaluated to be: 0.0807, 0.0822, 0.0840 and 0.0850, respectively. Therefore, the performance of the supervised plant under specifications #1, #2 and #3 is inferior, similar, and superior, respectively, to that of the unsupervised plant. As expected, the optimal supervisor has better performance than that of Supervisor #3. Notice that Supervisor #3 does disable the controllable event k from the state 3 to state 4 and the optimal supervisor does not. That is, the optimal supervisor allows continuing operation of an unhealthy engine while the remaining engine is in good condition.

8. Summary, conclusions, and future research

This paper reviews the concept, formulation and validation of a signed real measure for regular languages and their sublanguages based on the principles of automata theory and real analysis. While the domain of the measure μ, i.e., $2^{L(G_i)}$, is partially ordered, its range, which is a subset of $\mathbf{R} \equiv (-\infty, \infty)$, becomes totally ordered. As a result, the relative performance of different supervisors can be quantitatively evaluated in terms of the real signed measure of the supervised sublanguages. Positive weights are assigned to good marked states and negative weights to bad marked states so that a controllable supervisor is rewarded (penalized) for deleting strings terminating at bad (good) marked states. In order to evaluate and compare the performance of different supervisors a common quantitative tool is required. To this effect, the proposed procedure computes the measure of the supervised sublanguage generated by a supervisor using the event cost and characteristic function assigned for the unsupervised plant. Cost assignment to each event based on the state, where it is generated, has been shown similar to the conditional probability of the event. On the other hand, the characteristic function is chosen based on the designer's perception of the individual state's impact on the system performance. Two techniques are presented to compute the language measure for a DFSA. One of them yields a closed form solution that is obtained as the unique solution of a set of linearly independent algebraic equations. The other is based on a recursive procedure. The computational complexity of both language measure algorithms is identical and is of polynomial order in the number of states of the DFSA. As such it is relatively straight-forward to develop software analysis tools in standard languages such as Matlab, C, and Java (Wang *et al.* 2003).

8.1. *Recommendations for future research*

Optimal discrete-event supervisory control can be enhanced through appropriate usage of the language measure. For example, in the current configuration of the optimal control algorithm (Ray *et al.* 2004), if there is no state q_j such that $\mu_j^0 < 0$, then the plant performance cannot be improved by event disabling and the null controller S^0 (i.e., no disabled event) is the optimal controller for the given plant. This restriction can be lifted through an appropriate performance index that would be a function of the language measure $\boldsymbol{\mu}$ but not identically equal to $\boldsymbol{\mu}$ as reported in Ray *et al.* (2004). Work in this direction is in progress and is expected to be reported in a forthcoming publication.

Synthesis of supervisory control systems may become a significant challenge if some of the events are delayed, intermittent, or not observable at all, possibly due to sensor faults or malfunctions in network communication links. In that case, the control algorithms may turn out to be computationally very complex because of delayed or lost information on the plant dynamics. Future work in this direction should involve research on construction of language measures under partial observation (Chattopadhyay and Ray 2004) and synthesis of optimal control policies under partial observation to mitigate the detrimental effects of loss of observability. The latter research could be an extension of the earlier work on optimal control under full observation (Ray *et al.* 2004).

It would be a challenging task to extend the concept of (regular) language measure for languages higher up in the Chomsky Hierarchy (Martin 1997) such as context free and context sensitive languages. This extension would lead to controller synthesis when the plant dynamics is modelled by non-regular languages such as the Petri-Net. The research thrust should focus on retaining the polynomial order of computational complexity.

Another critical issue is how to extend the language measure for timed automaton, especially if the events are observed with varying delays at different states. Another research topic that may also be worth investigating is: how to extend the field $GF(2)$, over which the vector space of languages has been defined, to richer fields like the set of real numbers.

Other areas of future research include applications of the language measure in anomaly detection, model identification, model order reduction, and analysis and synthesis of interfaces between the continuously-varying and discrete-event spaces in the language-measure setting.

Appendices A: Measure theory

This appendix introduces the concepts of standard measure-theoretic quantities that are used to establish the language measure in the main body of this paper.

Definition A.1: A σ-algebra M of a nonempty language $L(G_i) \subseteq \Sigma^*$ is a collection of subsets of $L(G_i)$ which satisfies the following three conditions.

(i) $L(G_i) \in M$;
(ii) If $K \in M$, then $(L(G_i) - K) \in M$;
(iii) $\cup_{j=1}^{\infty} K_j \in M$ if $K_j \in M \ \forall j$.

Definition A.2: An at most countable collection $\{L_k\}$ of members of a σ-algebra M is a partition of a member $L \in M$ if $L = \cup_k L_k$ and $L_k \cap L_j = \emptyset \ \forall k \neq j$.

Definition A.3: Let M be a σ-algebra of $L(G_i)$. Then, the set function μ: $M \rightarrow \mathbf{R} \equiv (-\infty, +\infty)$, is called a signed real measure if the following two conditions are satisfied (Rudin 1987):

(i) $\mu(\emptyset) = 0$;
(ii) $\mu(\cup_{j=1}^{\infty} L_j) = \sum_{j=1}^{\infty} \mu(L_j)$ for every partition $\{L_j\}$ on any member $L \in M$.

Note that, unlike a positive measure (e.g., the Lebesgue measure), μ is finite such that the series in part (ii) of Definition A.3 converges absolutely in $\mathbf{R}$ and the result is independent of any permutation of the terms under union.

Definition A.4: Relative to the signed real measure μ, a sublanguage $L \in M$ is defined to be

(i) null, denoted as $L = 0$, if $\mu(L \cap J) = 0, \forall J \in M$;
(ii) positive, denoted as $L > 0$, if $L \neq 0$ and $\mu(L \cap J) \geq 0, \forall J \in M$;
(iii) negative, denoted as L, if $L \neq 0$ and $\mu(L \cap J) \leq 0$, $\forall J \in M$.

Definition A.5: Total variation $|\mu|$ on a σ-algebra M is defined as

$$|\mu|(L) = \sup \sum_k |\mu(L_k)| \tag{52}$$

$\forall L \in M$ where the supremum is taken over all partitions $\{L_k\}$ of L.

Proposition A.1: *Total variation measure $|\mu|$ of any regular language L is non-negative and finite i.e., $|\mu|(L) \in [0,\infty)$. The proof follows from standard theorems on complex measures (Rudin 1987).*

Total variation can be, in general, defined for complex measures (Rudin 1987) but it is restricted to a signed real measure in this paper. The total variation of a real signed measure μ, can be represented as $|\mu| = \mu^+ + \mu^-$ where μ^+ and μ^- are called positive and negative variations of μ and are defined as

$$\mu^+ = \tfrac{1}{2}(|\mu| + \mu) \qquad \text{and} \qquad \mu^- = \tfrac{1}{2}(|\mu| - \mu). \tag{53}$$

Both μ^+ and μ^- are positive measures on M. It also follows from the above equation that $\mu = \mu^+ - \mu^-$. This representation of μ as the difference of positive measure μ^+ and μ^- is known as the Jordan Decomposition of μ (Rudin 1987).

Proposition A.2: *Every sublanguage $L \in M$ can be partitioned as $L = L^0 \cup L^+ \cup L^-$ where the mutually exclusive sublanguages L^0, L^+ and L^- are called null, positive, and negative, respectively, relative to a signed real measure μ.*

Proof: The proof is based on the Hahn Decomposition Theorem (Rudin 1987). □

As a consequence of the above result, the following relations hold $\forall L \in M$ for positive and negative variations:

$$\mu^+(L) = \mu\left(L \bigcap L^+\right) \quad \text{and} \quad \mu^-(L) = -\mu\left(L \bigcap L^-\right). \tag{54}$$

Appendix B: Convergence of the language measure

This appendix establishes necessary and sufficient conditions for finiteness of the measure μ, based on certain properties of non-negative matrices, which are stated without proof. Details of these results are available in (Senata 1973, Plemmons and Berman 1979).

Definition B.1: Let A and B be real square matrices of the same order n. Then, the notations for inequalities are as follows:

$$\begin{aligned} A \geq B &\quad \text{if } a_{ij} \geq b_{ij},\ \forall i,j \\ A > B &\quad \text{if } A \geq B,\ A \neq B \\ A \gg B &\quad \text{if } a_{ij} > b_{ij},\ \forall i,j. \end{aligned}$$

If the matrix A satisfies the condition $A > \mathbf{0}$, i.e. the null matrix, then A is called a non-negative matrix and if the condition $A \gg 0$ is satisfied, then it is called a positive matrix.

Definition B.2: A square matrix A of order n is cogradient to a matrix E if $PAP^T = E$ for a permutation matrix P; and A is called reducible if A is cogradient to

$$E = \begin{bmatrix} B & \mathbf{0} \\ C & D \end{bmatrix},$$

where B and C are square matrices, or if $n = 1$ and $A = \mathbf{0}$. Otherwise, A is irreducible.

It follows from the above definition that a positive matrix is always irreducible.

Proposition B.1: *A non-negative matrix A is irreducible if and only if, for every (i,j) there exists a natural number k such that $a_{ij}^{(k)} > 0$, where $a_{ij}^{(k)}$ denotes the (i,j)th element of A^k.*

Proposition B.2: *If $A \geq \mathbf{0}$ is irreducible and $B \geq \mathbf{0}$, then $A + B$ is irreducible.*

Another characterization of irreducibility of a non-negative square matrix has a graph-theoretic interpretation. This relationship can help to determine under what conditions a given finite state automaton G, which represents the supervised or unsupervised plant model is irreducible by looking at connectivity of its states. The following definitions are needed to arrive at this conclusion.

Definition B.3: The associated directed graph, $G(A)$ of a square matrix A of order n, consists of n vertices $P_1, P_2, \ldots, P_n$ where an edge leads from P_i to P_j if and only if $a_{ij} \neq 0$.

Definition B.4: A directed graph G is strongly connected if for any ordered pair (P_i, P_j) of vertices of G, there exists a sequence of edges which leads from P_i to P_j.

Proposition B.3: *Given a matrix A, it is irreducible if and only if $G(A)$ is strongly connected.*

If A is a non-negative square matrix, then the following relationship holds between the spectral radius (i.e., maximum absolute eigenvalue) ρ of non-negative matrices.

Proposition B.4: *If $\mathbf{0} \leq A \leq B$ and $A + B$ is irreducible then $\rho(A) < \rho(B)$.*

Definition B.5: A square matrix S of order n is called (row) stochastic if it satisfies

$$s_{ij} \geq 0, \quad \sum_{j=1}^{n} s_{ij} = 1, \quad 1 \leq i \leq n. \tag{55}$$

Proposition B.5: *The maximum eigenvalue of a stochastic matrix S is one, i.e.* $\rho(S) = 1$. *A non-negative matrix A is stochastic if and only if e is an eigenvector of A corresponding to the eigenvalue one, where e is the vector all of whose entries are equal to one.*

In order to show that $(I - \mathbf{\Pi})^{-1}$ is invertible it suffices to show that $\rho(\mathbf{\Pi}) < 1$.

Theorem B.1: *If* $\rho(\mathbf{\Pi}) < 1$ *then there exists at least one* i, $1 \leq i \leq n$, *such that* $\sum_{j=1}^{n} \pi_{ij} < 1$.

Proof: Proof follows from the fact that if $\sum_{j=1}^{n} \pi_{ij} = 1$ $\forall i$, then $\mathbf{\Pi}$ would be a stochastic matrix by Definition B.5. Hence, by Proposition, B.5 $\rho(\mathbf{\Pi}) = 1 \Rightarrow (I - \mathbf{\Pi})^{-1}$ is not invertible. □

Theorem B.2: *If* $\sum_{j=1}^{n} \pi_{ij} < 1$ $\forall i$, $1 \leq i \leq n$, *then* $\rho(\mathbf{\Pi})$.

Proof: Let $\theta_i = (1 - \sum_{j=1}^{n} \pi_{ij})/n > 0$. Let S be a matrix of order n which is defined in the following manner

$$s_{ij} = \theta_i + \pi_{ij}, \quad \forall 1 \leq i, j \leq n.$$

It is clear that $S \gg \mathbf{0}$ and hence S is irreducible. Also S is a stochastic matrix and by Proposition B.5, the spectral radius $\rho(S) = 1$. Since $\mathbf{0} \leq \mathbf{\Pi} <$ and $\mathbf{\Pi} + S$ is irreducible by Proposition B.2, it follows that $\rho(\mathbf{\Pi}) < \rho(S) = 1$ from Proposition B.4. □

The above sufficiency condition is more strict than the necessary condition required in Theorem B.1. However, the necessary condition is not sufficient as seen from the following example.

$$\mathbf{\Pi} = \begin{pmatrix} 0.2 & 0 & 0.8 & 0 \\ 0 & 0.2 & 0.3 & 0.5 \\ 0.5 & 0 & 0.5 & 0 \\ 0.1 & 0.2 & 0.4 & 0 \end{pmatrix}.$$

This matrix $\mathbf{\Pi}$ satisfies conditions as required in Theorem B.1, but $\rho(\mathbf{\Pi}) = 1$. It is possible to relax the strict inequality $\sum_{j=1}^{n} \pi_{ij} < 1$ $\forall i$, $1 \leq i \leq n$ in Theorem B.2, but with additional conditions on structure of $\mathbf{\Pi}$. For example, under such relaxed conditions, if $\mathbf{\Pi} + S$ is irreducible, then still $\rho(\mathbf{\Pi}) < 1$. This follows from the fact that application of Proposition B.4, only requires irreducibility of $\mathbf{\Pi} + S$. In order to determine the irreducibility of a matrix, the graph-theoretic interpretation, described earlier, can be a useful tool.

Acknowledgements

The author wishes to thank his former and current students and his colleagues for their contributions in the development of the language measure concept. The author specifically acknowledges the technical contributions of Mr. Amit Surana, Mr. Ishanu Chattopadhyay, Dr. Xi Wang, Dr. Jinbo Fu, and Professor Constantino Lagoa in the development of the language measure theory and its usage in discrete event supervisory control.

This work has been supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office (ARO) under Grant No. DAAD19-01-1-0646; and NASA Glenn Research Center under Grant No. NNC04GA49G.

References

C.G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, Boston, MA: Kluwer Academic, 1999.

I. Chattopadhyay and A. Ray, "A language measure for partially observed discrete-event supervisory control systems", in *Proceedings of 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, Atlantis, December 2004, pp. 45–50.

Y.-C. Ho and X.-R. Cao, *Perturbation Analysis of Discrete Event Dynamic Systems*, Boston, MA: Kluwer Academic, 1991.

J.E. Hopcroft, R. Motwani and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 2nd edn, Boston, MA: Addison-Wesley, 2001.

R.E. Kalman, P.L. Falb and M.A. Arbib, *Topics in Mathematical System Theory*, Boston, MA: McGraw-Hill, 1969.

R. Kumar and V. Garg, *Modeling and Control of Logical Discrete Event Systems*, Boston, MA: Kluwer Academic, 1995.

A.M. Law and W.D. Kelton, *Simulation Modeling and Analysis*, 2nd edn, New York, NY: McGraw-Hill International, 1991.

J.C. Martin, *Introduction to Languages and the Theory of Computation*, 2nd edn, Boston, MA: McGraw-Hill, 1997.

A.W. Naylor and G.R. Sell, *Linear Operator Theory in Engineering and Science*, New York, NY: Springer-Verlag, 1982.

S. Phoha, E. Peluso and R.L. Culver, "A high fidelity ocean sampling mobile network (samon) simulator", *IEEE Journal of Oceanic Engineering, Special Issue on Autonomous Ocean Sampling Networks*, 26, pp. 646–653, 2002.

R.J. Plemmons and A. Berman, *Nonnegative Matrices in the Mathematical Science*, New York: Academic Press, 1979.

P.J. Ramadge and W.M. Wonham, "Supervisory control of a class of discrete event processes", *SIAM J. Control and Optimization*, 25, pp. 206–230, 1987.

A. Ray, J. Fu and C. Lagoa, "Optimal supervisory control of finite state automata", *International Journal of Control*, 77, pp. 1083–1100, 2004.

A. Ray and S. Phoha, "Signed real measure of regular languages for discrete-event automata, *International Journal of Control*, 76, pp. 1800–1808, 2003.

W. Rudin, *Real and Complex Analysis*, 3rd edn, New York: McGraw-Hill, 1987.

E. Senata, *Non-negative Matrices*, New York: John Wiley, 1973.

X. Wang and A. Ray, "A language measure for performance evaluation of discrete-event supervisory control systems", *Applied Mathematical Modelling*, 28, pp. 817–833, 2004.

X. Wang, A. Ray and A. Khatkhate, "On-line identification of language measure parameters for discrete event supervisory control", *Applied Mathematical Modelling*, 29, pp. 597–613, 2005.

X. Wang, A. Ray, S. Phoha and J. Liu, "J-des: a graphical interactive package for analysis and synthesis of discrete event systems", in *Proceedings of American Control Conference*, Denver, Colorado, June 2003, pp. 3405–3410.

S. Yu, *Regular Languages, Handbook of Language Theory*, Chapter 2, Vol. 1, Berlin: Springer, 1997.

International Journal of Control
Vol. ??, No. ?, Month?? 2007, 1–20

Language-measure-theoretic optimal control of probabilistic finite-state systems

I. CHATTOPADHYAY and A. RAY*

Mechanical Engineering Department, The Pennsylvania State University,
University Park, PA 16802, USA

(*Received 10 October 2006; in final form 17 February 2007*)

Supervisory control theory for discrete event systems, introduced by Ramadge and Wonham, is based on a non-probabilistic formal language framework. However, models for physical processes inherently involve modelling errors and noise-corrupted observations, implying that any practical finite-state approximation would require consideration of event occurrence probabilities. Building on the concept of signed real measure of regular languages, this paper formulates a comprehensive theory for optimal control of finite-state probabilistic processes. It is shown that the resulting discrete-event supervisor is optimal in the sense of elementwise maximizing the renormalized langauge measure vector for the controlled plant behaviour and is efficiently computable. The theoretical results are validated through several examples including the simulation of an engineering problem.

1. Introduction

Supervisory control theory (SCT) of discrete-event systems (DES), pioneered by Ramadge and Wonham (1987), models a physical or human-engineered process as a finite-state language generator and constructs a supervisor that attempts to constrain the ‘‘supervized’’ plant behaviour within a specification language. The original theory is based on a deterministic language framework. Although allowing non-determinism in the sense that more than one continuation of a generated event trace (i.e., a string) is possible, no attempt is made to quantify this randomness. As Wonham himself observes in Lawford and Wonham (1993), ‘‘the choice of a possible continuation of a string is made by some internal structure unmodeled by the systems designer’’. The notion of probabilistic languages in the context of studying qualitative stochastic behaviour of discrete-event systems first appears in Garg (1992a, b), where the concept of p-languages (‘p’ implying probabilistic) is introduced and an algebra is developed to model probabilistic languages based on concurrency (Milner 1989). A regular p-language is essentially a set of prefix-closed traces of events, generated by a finite-state automaton with probabilities associated with the transitions. A p-language-theoretic model differs in several important aspects from other discrete-event models of stochastic analysis such as Markov chains (Cassandras and Lafortune 1999), stochastic Petri nets (Molloy 1982, Chung *et al.* 1994), probabilistic automata (Rabin 1963, Paz 1971, Doberkat 1981), and fuzzy models (Lee and Zadeh 1969). Garg *et al.* (1999) and Kumar and Garg (2001) provide a brief comparison of the p-language-theoretic modelling paradigm with the above-mentioned theories.

Lawford and Wonham (1993) have attempted to extend discrete-event (SCT) to plants modelled by p-languages, where a formal statement of the probabilistic supervisory control problem (PSCP) first appears and the notion of probabilistic supervision is introduced by random disabling of controllable events. The key difference from other stochastic supervision approaches (e.g., Mortzavian 1993) lies in the fact that the computed probabilistic supervisor is not allowed to change the

*Corresponding author. Email: axr2@psu.edu

International Journal of Control
ISSN 0020–7179 print/ISSN 1366–5820 online © 2007 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00207170701286322

underlying plant dynamics in the following sense: "The probabilistic effect of random disablement is determined entirely by the plant". The control objective is specified as a p-language and necessary and sufficient conditions are derived for existence of a probabilistic supervisor that attempts to restrict the plant language within the control specification in a probabilistic sense. The theory of supervision of p-languages is further developed by Kumar and Garg (2001), where the control objective is specified as upper and lower bound constraints. The upper bound is a non-probabilistic language that serves as a legality constraint, while the lower bound is a p-language. This relatively relaxed approach to control objective specification allows for a non-probabilistic supervisor that attempts to cut down illegal event traces, while ensuring that legal traces occur with probabilities greater than or equal to that specified by the lower bound. Intuitively, the designed supervisor stops "bad" strings from occurring while guaranteeing that "good" strings occur with some minimum pre-set frequency. However, construction of such a control objective specification may not be possible in many applications (e.g., battlefield command, control, communications, and intelligence (C^3I) (Phoha *et al.* 2002)), especially if the decisions are to made in real time. For the theory to be useful in practice, one must generate the specification from the definition of the physical problem at hand. Given that one has to come up with a non-probabilistic language to serve as the upper legality constraint and a probabilistic language to serve as the lower bound, this goal may not always be achievable. The situation becomes worse for non-stationary stochastic environments, where the control specifications may have to be updated online.

A significantly simplified approach to the above problem is reported by Ray (2005) and Ray *et al.* (2005), where the control objective is specified as characteristic weights on the states of the plant automaton. These weights are normalized in the interval $[-1, 1]$ with positive weights assigned to good states and negative weights to bad states. A signed real measure of regular languages (of event traces) is defined as a function of the characteristic weights and the state transition probabilities; and supervisory control laws are synthesized by elementwise maximizing the language measure vector (Ray *et al.* 2004, 2005). Intuitively, the supervisor ensures that the generated event traces cause the plant to visit the "good" states while attempting to avoid the "bad" states in a probabilistic sense. As mentioned earlier, Kumar and Garg's work on supervisory control of probabilistic automata (Kumar and Garg 2001) also has a notion of "good" and "bad" strings. However, the classification is strictly binary; the theory has no way of saying if one "good" string is "better" than another "good" string and *vice versa*. This implies that the supervisor must eliminate all bad strings and hence may not be optimal, or fail to exist if the conditions defined in Kumar and Garg (2001) are not satisfied. In contrast, in the measure-theoretic approach (Ray 2005), the event traces are less or more desirable in a continuous scale with the supervisor optimizing the controlled plant behaviour to ensure that the "most" desirable strings occur "most" often. This has an immediate advantage that the problem of existence disappears; the optimal supervisor always exists and can be computed effectively with polynomial complexity. The latter approach is, in one sense, closer to Markov chain modelling since the control specification is state-based. However, as shown by Ray (2005), this does not restrict the modeling power of the technique. It is shown in Kumar and Garg (2001) that, in general, maximally permissive supervisors are non-unique. For the measure-theoretic approach, however, the optimization is shown to yield unique maximal permissiveness among all optimal supervisors (Ray *et al.* 2004, 2005).

Optimal control in the context of discrete event dynamic systems has been addressed earlier by several investigators as cited in (Ray *et al.* 2004). For example, Sengupta and Lafortune (1998) have analysed non-probabilistic DES with assigned event and control costs; the optimal supervisor is computed in the framework of dynamic programming (DP) with two critical assumptions to guarantee polynomial complexity of the solution: all costs are strictly positive and there is only one marked state (Sengupta and Lafortune 1998, p. 34). The work reported in Ray (2005) and Ray *et al.* (2005) is different in the sense that the latter deals with probabilistic automata and the optimization, even in the completely general case, has guaranteed polynomial complexity of $O(n^3)$, where n is the number of states in the unsupervised plant model. The measure-theoretic approach was originally reported for a restricted class of terminating p-languages Ray (2005) and Ray *et al.* (2005); and this restriction has been eliminated in a subsequent publication (Chattopadhyay and Ray 2006a).

The notion of terminating and non-terminating automata is originally due to Garg (19992a, b). A probabilistic automaton is terminating if there exist states at which the sum of the probabilities of all defined events is strictly less than 1. The interpretation is that the difference of the sum from 1 is the probability that the plant terminates operation at that particular state. It is shown in Ray (2005) that the language measure vector can be expressed as $[\mathbb{I} - \boldsymbol{\Pi}]^{-1}\boldsymbol{\chi}$ where $\boldsymbol{\Pi}$ is the transition probability matrix and $\boldsymbol{\chi}$ is the characteristic vector, where π_{ij} is the probability of transition from the ith state to the jth state and χ_i is the characteristic weight of the state i). A sufficient condition for the inverse of $\left[\mathbb{I} - \Pi\right]$ to exist is that $\sum_j \pi_{ij} < 1 \ \in i$, i.e., the plant has a strictly non-zero probability of termination from each state.

This paper eliminates the above restrictive assumption by adopting the recently reported renormalized measure of regular languages (Chattopadhyay and Ray 2006a) as the performance index. It also extends the measure-theoretic concept for optimal control of terminating plants (Ray *et al.* 2004) to non-terminating plant models, which requires a minor modification of the control philosophy as explained below.

Supervisors in the SCT paradigm are allowed to affect the underlying plant behaviour by selectively disabling controllable events (Ramadge and Wonham 1987). In case of terminating p-languages, a similar approach suffices; the supervisor selectively nulls the occurrence probability of controllable events to achieve the desired control objective. However, the non-terminating case poses a problem since any such disabling action converts the system to a terminating p-language (i.e., the probabilities of the events defined at a state fail to add up to 1). The solution (Kumar and Garg 2001) is to proportionately increase the probabilities of the remaining enabled events at the state at which event disabling is undertaken. An alternative approach is proposed in this paper, where each disabled event creates a self loop at the state (at which the event was generated) with occurrence probability of the original transition.

The paper is organized in six sections and an appendix. Section 2 lays down the basic framework of the analysis and briefly reviews the original notion of language measure (Ray 2005) and its renormalization (Chattopadhyay and Ray 2006a). Section 3 formulates the optimal control problem based on the concept of renormalized measure and presents the key results. Section 4 presents a solution of the optimal control problem and derives the necessary algorithms for its implementation. Section 5 presents an engineering example, where the optimal supervisor is designed for a three-processor message decoding system. The paper is summarized and concluded in § 6 along with recommendations for future research. Appendix G establishes bounds on the derivatives of the renormalized measure that is necessary for formulation of the optimal control law in § 3.

2. Preliminary concepts

This section briefly reviews the concept of signed real measure of regular languages Ray (2005) and Ray *et al.* (2005) followed by a review of the notion of renormalized measure and the pertinent notations used in the sequel.

2.1 *Brief review of language measure*

Let the plant behaviour be modelled as a deterministic finite state automaton (DFSA) as $G_i \triangleq (Q, \Sigma, \delta, q_i, Q_m)$, where Q is the finite set of states with $|Q| = n$, and $q_i \in Q$ is the initial state; Σ is the (finite) alphabet of events with $|\Sigma| = m$; the Kleene closure of Σ is denoted as $\Sigma^\star$ that is the set of all finite-length strings of events including the empty string ε; the (possibly partial) function $\delta: Q \times \Sigma \to Q$ represents state transitions and $\delta^\star: Q \times \Sigma^\star \to Q$ is an extension of δ; and $Q_m \subseteq Q$ is the set of marked (i.e., accepted) states.

Definition 1: The language $L(G_i)$ generated by the *DFSA* G_i is defined as $L(G_i) = \{s \in \Sigma^* \mid \delta^*(q_i, s) \in Q\}$.

Definition 2: The marked language $L_m(G_i)$ by the *DFSA* G_i is defined as $L_m(G_i) = \{s \in \Sigma^* \mid \delta^*(q_i, s) \in Q_m\}$.

The language $L(G_i)$ of the DFSA G_i is partitioned as the non-marked and the marked languages, $L^o(G_i) \triangleq L(G_i) - L_m(G_i)$ and $L_m(G_i)$, consisting of event strings that, starting from $q_i \in Q$, terminate at one of the non-marked states in $Q - Q_m$ and one of the marked states in Q_m, respectively. The set Q_m is partitioned into Q_m^+ and Q_m^- where Q_m^+ contains all good marked states that one may desire to terminate on, and Q_m^- contains all bad marked states that one would attempt to avoid terminating on, although it may not always be possible to bypass a bad state before reaching a good state. The marked language $L_m(G_i)$ is further partitioned into $L_m^+(G_i)$ and $L_m^-(G_i)$ consisting of good and bad strings that, starting from q_i, terminate on Q_m^+ and Q_m^-, respectively.

A signed real measure $\mu: 2^{\Sigma^\star} \to R \triangleq (-\infty, \infty)$ is constructed for quantitative evaluation of every event string $s \in \Sigma^\star$. The language $L(G_i)$ is decomposed into null, i.e., $L^o(G_i)$, positive, i.e., $L_m^+(G_i)$, and negative, i.e., $L_m^-(G_i)$ sublanguages.

Definition 3: The language of all strings that, starting at a state $q_i \in Q$, terminates on a state $q_j \in Q$, is denoted as $L(q_i, q_j)$. That is,

$$L(q_i, q_j) \triangleq \{s \in L(G_i) : \delta^\star(q_i, s) = q_j\}. \tag{1}$$

Definition 4: The characteristic function that assigns a signed real weight to each state $q_i, i = 1, 2, \ldots, n$, is defined as: $\chi: Q \to [-1, \ 1]$ such that

$$\chi(q_j) \in \begin{cases} [-1, \ 0) & \text{if } q_j \in Q_m^- \\ \{0\} & \text{if } q_j \notin Q_m \\ (0, \ 1] & \text{if } q_j \in Q_m^+ \end{cases}$$

Definition 5: The event cost is conditioned on a *DFSA* state at which the event is generated, and is defined as $\tilde{\pi} : \Sigma^{\star} \times Q \to [0, 1]$ such that $\forall q_j \in Q, \forall \sigma_k \in \Sigma$, $\forall s \in \Sigma^{\star}$,

1. $\tilde{\pi}[\sigma_k, q_j] \triangleq \tilde{\pi}_{jk} \in [0, 1)$; $\sum_k \tilde{\pi}_{jk} < 1$;
2. $\tilde{\pi}[\sigma, q_j] = 0$ if $\delta(q_j, \sigma)$ is undefined; $\tilde{\pi}[\epsilon, q_j] = 1$;
3. $\tilde{\pi}[\sigma_k s, q_j] = \tilde{\pi}[\sigma_k, q_j]\ \tilde{\pi}[s, \delta(q_j, \sigma_k)]$.

The event cost matrix, denoted as $\widetilde{\boldsymbol{\Pi}}$-matrix, is defined as

$$\widetilde{\Pi} = \begin{bmatrix} \tilde{\pi}_{11} & \dots & \tilde{\pi}_{1m} \\ \vdots & \ddots & \vdots \\ \tilde{\pi}_{n1} & \dots & \tilde{\pi}_{nm} \end{bmatrix}$$

An application of the induction principle to part (3) in Definition 5 shows $\tilde{\pi}[st, q_j] = \tilde{\pi}[s, q_j]\tilde{\pi}[t, \delta^{*}(q_j, s)]$. The condition $\Sigma_k \tilde{\pi}_{jk} < 1$ provides a sufficient condition for the existence of the real signed measure as discussed in Ray (2005) along with additional comments on the physical interpretation of the event cost.

Now let us define the measure of a sublanguage of the plant language $L(G_i)$ in terms of the signed characteristic function χ and the non-negative event cost $\tilde{\pi}$.

Definition 6: The signed real measure μ of a singleton string set $\{s\} \subseteq L(q_i, q_j) \subseteq L(G_i) \in 2^{\Sigma^{\star}}$ is defined as

$$\mu(\{s\}) \triangleq \tilde{\pi}(s, q_i)\chi(q_j) \qquad \forall s \in L(q_i, q_j).$$

The signed real measure of $L(q_i, q_j)$ is defined as

$$\mu\big(L(q_i, q_j)\big) \triangleq \sum_{s \in L(q_i, q_j)} \mu(\{s\})$$

and the signed real measure of a *DFSA* G_i, initialized at the state $q_i \in Q$, is denoted as

$$\mu_i \triangleq \mu(L(G_i)) = \sum_j \mu\big(L(q_i, q_j)\big).$$

Definition 7: The state transition cost of the *DFSA* is defined as a function π: $Q \times Q \to [0, 1)$ such that

$$\pi(q_j, q_k) = \begin{cases} 0 & \text{if } \{\sigma \in \Sigma\colon \delta(q_j, \sigma) = q_k\} = \emptyset \\ \sum_{\sigma \in \Sigma\colon \delta(q_j, \sigma) = q_k} \tilde{\pi}(\sigma, q_j) \triangleq \pi_{jk} & \text{otherwise} \end{cases}. \tag{2}$$

The state transition cost matrix, denoted as $\boldsymbol{\Pi}$-matrix, is defined as

$$\boldsymbol{\Pi} = \begin{bmatrix} \pi_{11} & \dots & \pi_{1n} \\ \vdots & \ddots & \vdots \\ \pi_{n1} & \dots & \pi_{nn} \end{bmatrix}.$$

It has been shown in (Ray 2005 and Ray *et al.* 2005) that the measure $\mu_i \triangleq \mu(L(G_i))$ of the language $L(G_i)$, with the initial state q_i, can be expressed as $\mu_i = \sum_j \pi_{ij} \mu_j + \chi_i$ where $\chi_i \triangleq \chi(q_i)$. Equivalently, in vector notation

$$\boldsymbol{\mu} = \boldsymbol{\Pi}\boldsymbol{\mu} + \boldsymbol{\chi} \Longrightarrow \boldsymbol{\mu} = [\mathbb{I} - \boldsymbol{\Pi}]^{-1}\boldsymbol{\chi}, \tag{3}$$

where the measure vector $\boldsymbol{\mu} \triangleq [\mu_1\ \mu_2\ \cdots\ \mu_n]^T$ and the characteristic vector $\boldsymbol{\chi} \triangleq [\chi_1\ \chi_2\ \cdots\ \chi_n]^T$; and the condition $\sum_j \tilde{\pi}_{ij} < 1\ \ \in i$ (see Definition 5) is sufficient for the inverse to exist.

Although the preceding analysis reported in (Ray 2005 and Ray *et al.* 2005) was intended for non-probabilistic regular languages, the event costs can be easily interpreted as occurrence probabilities. As such the $\widetilde{\boldsymbol{\Pi}}$-matrix is analogous to the morph matrix of a Markov chain in the sense that an element $\tilde{\pi}_{ij}$ represents the probability of the *j*th event occurring at the *i*th state with the exception that the strict inequality condition $\sum_j \tilde{\pi}_{ij} < 1$ is enforced instead of satisfying the equality. Equivalently, the $\boldsymbol{\Pi}$-matrix is analogous to the state transition probability matrix of a Markov chain in the sense that an element π_{jk} is analogous to the transition probability from state q_j to state q_k with the exception that the strict inequality condition $\sum_k \pi_{jk} < 1$ is enforced instead of satisfying the equality. It follows that the preceding analysis is applicable to the case of terminating probabilistic languages (Garg *et al.* 1992a, b) that have a strictly non-zero probability of termination at each state.

Let Σ^u denote the set of all unmodelled events in the terminating plant. A new unmarked absorbing state q_{n+1}, called the dump state (Ramadge and Wonham 1987), is created and the transition function δ is extended to $\delta_{\text{ext}} : (Q \cup \{q_{n+1}\}) \times (\Sigma \cup \Sigma^u) \to (Q \cup \{q_{n+1}\})$. The residue $\theta_j = 1 - \sum_k \pi_{jk}$ denotes the probability of transition from q_j to q_{n+1}. Consequently, the $\boldsymbol{\Pi}$-matrix (see Definition 7) is augmented to obtain the stochastic state transition probability matrix as

$$\boldsymbol{\Pi}_{\text{aug}} = \begin{bmatrix} \pi_{11} & \pi_{12} & \dots & \pi_{1n} & \theta_1 \\ \pi_{21} & \pi_{22} & \dots & \pi_{2n} & \theta_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \pi_{n1} & \pi_{n2} & \dots & \pi_{nn} & \theta_n \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}. \tag{4}$$

Since the dump state q_{n+1} is not marked (Ramadge and Wonham 1987), it follows from Definition 4 that the corresponding state weight $\chi_{n+1} = 0$. Hence, the χ-vector is augmented as

$$\boldsymbol{\chi}_{\text{aug}} = [\boldsymbol{\chi}^T\ 0]^T. \tag{5}$$

Denoting $\boldsymbol{\Theta} = [\theta_1\ \theta_2 \cdots \theta_n]^T$, where $\theta_i \in (0, 1)$ is the probability of transition from the state q_i to the dump state, it follows from equations (4) and (5) that the measure of the augmented system (Chattopadhyay and Ray 2006a) is

$$\boldsymbol{\mu}_{\text{aug}}(\boldsymbol{\Theta}) = [\boldsymbol{\mu}(\boldsymbol{\Theta})^T\ 0]^T. \tag{6}$$

Then, the event cost matrix (see Definition 5) and the state transition cost matrix (see Definition 7) can be represented as

$$\widetilde{\boldsymbol{\Pi}}(\boldsymbol{\Theta}) = \Big[\mathbb{I} - \text{Diag}[\boldsymbol{\Theta}]\Big]\widetilde{\mathbf{P}} \quad \text{and} \quad \boldsymbol{\Pi}(\boldsymbol{\Theta}) = \Big[\mathbb{I} - \text{Diag}[\boldsymbol{\Theta}]\Big]\mathbf{P}, \tag{7}$$

where $\widetilde{\mathbf{P}}$ and $\mathbf{P}$ are both stochastic matrices (Bapat and Raghavan 1997), i.e., $\Sigma_j \widetilde{\mathbf{P}}_{ij} = 1\ \forall i \in \{1, \ldots, m\}$ and $\Sigma_j \mathbf{P}_{ij} = 1\ \forall i \in \{1, \ldots, n\}$.

If the probability of termination (or equivalently the probability of transition to the dump state) is equal for all states, $q_i \in Q$, i.e., $\theta_i = \theta\ \ \forall i \in \{1, 2, \ldots, n\}$, then equation (6) is expressed as

$$\boldsymbol{\mu}_{\text{aug}}(\theta) = [\boldsymbol{\mu}(\theta)^T\ 0]^T \tag{8}$$

Consequently, $\widetilde{\boldsymbol{\Pi}}$ and $\boldsymbol{\Pi}$ in equation (7) are represented as

$$\widetilde{\boldsymbol{\Pi}}(\theta) = (1 - \theta)\widetilde{\mathbf{P}} \quad \text{and} \quad \boldsymbol{\Pi}(\theta) = (1 - \theta)\mathbf{P} \tag{9}$$

where θ is the uniform probability of termination at all states; and both $\widetilde{\mathbf{P}}$ and $\mathbf{P}$ retain the properties of stochastic matrices (Bapat and Raghavan 1997).

2.2 *Renormalization of language measure*

The notion of language measure has been recently extended to non-terminating models by assuming a uniform non-zero probability of termination (θ) at each state, renormalizing the language measure vector with respect to the probability of termination and computing the limit of the renormalized measure (Chattopadhyay and Ray 2006a) as $\theta \to 0^+$. As the probability of termination approaches zero (i.e., $\theta \to 0^+$), and the plant coincides with the desired non-terminating model in the limit. The construction of renormalized measure is briefly outlined below.

The regular language generated by the *DFSA* under consideration is a sublanguage of the Kleene closure Σ^* of the alphabet Σ, for which the automaton states can be merged into a single state. In that case, the state transition cost matrix $\boldsymbol{\Pi}(\theta)$ degenerates to the 1×1 matrix $[1 - \theta]$ and the normalized state weight vector $\boldsymbol{\chi}$ becomes one-dimensional and can be assigned as $\boldsymbol{\chi} = 1$; consequently, the measure vector $\boldsymbol{\mu}(\theta)$ degenerates to the scalar measure θ^{-1}. To alleviate the singularity of the matrix operator $[I - \boldsymbol{\Pi}(\theta)]$ at $\theta = 0$, the measure vector $\boldsymbol{\mu}(\theta)$ in (3) is normalized with respect to θ^{-1} to obtain the renormalized measure vector.

Definition 8: The renormalized measure is defined as

$$\boldsymbol{\nu}(\theta) = \theta\ \boldsymbol{\mu}(\theta) = \theta\ [\mathbb{I} - \boldsymbol{\Pi}(\theta)]^{-1}\boldsymbol{\chi} \tag{10}$$

and it follows from (8) that

$$\theta\ \boldsymbol{\mu}_{\text{aug}}(\theta) = [\boldsymbol{\nu}(\theta)^T\ 0]^T. \tag{11}$$

3. Optimal control problem: formulation

The following notations are needed for elementwise comparison of finite-dimensional vectors and matrices for the analysis developed in the sequel.

Notation 1: Let V^a and V^b be $(m \times n)$ real matrices. The following elementwise equality and inequalities imply that

$$\begin{aligned}
\left(V^a \equiv_{\mathbf{E}} V^b\right) &\Leftrightarrow \left(V^a_{ij} = V^b_{ij}\right) \forall i \in \{1, \ldots, n\}\ \forall j \in \{1, \ldots, m\} \\
\left(V^a \neq_{\mathbf{E}} V^b\right) &\Leftrightarrow \left(V^a_{ij} \neq V^b_{ij}\right) \exists\ i \in \{1, \ldots, n\},\ j \in \{1, \ldots, m\} \\
\left(V^a \geqq_{\mathbf{E}} V^b\right) &\Leftrightarrow \left(V^a_{ij} \geq V^b_{ij}\right) \forall i \in \{1, \ldots, n\}\ \forall j \in \{1, \ldots, m\} \\
\left(V^a >_{\mathbf{E}} V^b\right) &\Leftrightarrow \left(V^a_{ij} > V^b_{ij}\right) \forall i \in \{1, \ldots, n\}\ \forall j \in \{1, \ldots, m\}.
\end{aligned}$$

For the terminating plant, investigated in (Ray 2005 and Ray *et al.* 2005), the optimal supervisor selectively disables controllable transitions by setting their occurrence probabilities to zero. This implies that if $\boldsymbol{\Pi}^\star$ and $\boldsymbol{\Pi}$ are the transition probability matrices for the optimally supervised plant and the unsupervised plant, respectively, then

$$\boldsymbol{\Pi}^\star \leqq_{\mathbf{E}} \boldsymbol{\Pi}, \quad \text{i.e.,}\ \ \pi^\star_{ij} \leq \pi_{ij}.$$

Since for any non-trivial supervisor, there is at least one disabled transition in the supervised plant, i.e.,

$$\exists i, j \text{ such that } \pi_{i,j} > 0 \quad \text{and} \quad \pi^\star_{i,j} = 0$$

it follows that if the unsupervised plant is non-terminating, then any non-trivial supervision will result in a terminating model. The policy of Kumar and

Garg 2001 maintains the non-termination property by proportionately increasing the probabilities of the remaining enabled events at the state at which the disabling action is applied. The first issue here is that the supervisor must be able to affect the event occurrence probabilities, which is more than just inhibiting a transition. The second issue is that there is a possibility of disabling all events defined at a given state if all these events are controllable. In that case, the row sum cannot be maintained at 1 as it becomes strictly equal to zero. Thus, it is necessary to impose special constraints on the unsupervised plant to circumvent this situation. This paper investigates an alternative approach as described below.

Definition 9 (control philosophy)**:** Disabling any transition σ at a given state q results in reconfiguration of the automaton structure as: Set the self-loop $\delta(q,\sigma)=q$ with the occurrence probability of σ from the state q remaining unchanged in the supervised and unsupervised plants.

This is equivalent to adding a self-loop to the state at which the event is being disabled, with the same occurrence probability as the disabled transition.

Preposition 1: For the control philosophy in Definition 9, a supervised plant is non-terminating if and only if the unsupervised plant is non-terminating.

Proof: The proof follows from two lemmas.

Lemma 1: Each row sum of the $\mathbf{\Pi}$-matrix remains unchanged after supervisory actions for the control philosophy in Definition 9.

Proof: Let $\mathbf{\Pi}$ and $\mathbf{\Pi}^\dagger$ be the transition probability matrices for the unsupervised and supervised plants, respectively. Let there be exactly one disabled transition, in which a (controllable) event σ at state q_i is disabled and let the occurrence probability of σ at state q_i be $\tilde{\pi}$. If $\delta(q_i,\sigma)=q_k$, then it follows that

$$\mathbf{\Pi}^\dagger = \mathbf{\Pi} + \begin{array}{cc} \begin{matrix} & & & & k\text{th column} & \\ & & & & \downarrow & \end{matrix} & \\ \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 & 0 \\ \vdots & \ddots & & \vdots & \vdots & \vdots \\ 0 & \cdots & \tilde{\pi} & \cdots & -\tilde{\pi} & 0 \\ \vdots & \cdots & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix} & \begin{matrix} \\ \\ \leftarrow i\text{th row} \\ \\ \\ \\ \end{matrix} \end{array}$$

implying $\sum_j \pi_{ij}^\dagger = \sum_j \pi_{ij}\ \forall i$. The proof follows by induction on the number of disabled events. □

Lemma 2: Self-loops cannot be disabled.

Proof: For the control philosophy in Definition 9, disabling a self-loop at any given state causes regeneration of the self-loop at the same state with identical occurrence probability. □

It is evident from the above two lemmas that each row sum of the reconfigured $\mathbf{\Pi}$-matrix remains invariant. The proof of Proposition 1 is thus complete. □

Remark 1: The control philosophy in Definition 9 is natural in the following sense. If $q_i \rightarrow_\sigma q_k$, and the controllable event σ is disabled at state q_i, then the sole effect of the supervisory action is to prevent the plant from making a transition to the state q_k. That is, the plant is forced to stay at the original state q_i and this is represented by the additional self-loop at state q_i instead of the original arc from q_i to q_k.

The notion of controllability is now clarified in the context of the present paper.

Definition 10 (controllable transitions)**:** For a given plant, transitions that can be disabled in the sense of Definition 9 are defined to be controllable transitions in the sequel.

In view of Definition 10, controllability becomes state-based, i.e., transitions labelled by the same event may be controllable from one state and uncontrollable from some other state. This implies that the event alphabet Σ cannot be partitioned into uncontrollable and controllable events sets as proposed in Ramadge and Wonham (1987). Thus, a controllable transition $q_i \overset{\sigma}{\rightarrow} q_k$ refers to a triple $\{q_i,\sigma,q_k\}$ and the set of all such transitions is denoted by $\mathscr{C}$.

3.1 *Model specification*

Plant models considered in this paper are deterministic finite state automata (DFSA) with well-defined event occurrence probabilities. In other words, the occurrence of events is probabilistic, but the state at which the plant ends up, given a particular event has occurred, is deterministic. Furthermore, no emphasis is laid on the initial state of the plant and it is assumed that the plant may start from any state. Furthermore, having defined the characteristic state weight vector $\boldsymbol{\chi}$, it may not be necessary to specify the set of marked states, because if $\chi_i=0$, then q_i is not marked and if $\chi_i \neq 0$, then q_i is marked. Therefore, plant models with an arbitrary uniform termination probability $\theta \in (0,1)$, i.e., $\theta_i=\theta\ \ \forall i \in \{1,2,\ldots,n\}$, can be completely specified by a sextuple as

$$G(\theta) = \left(Q, \Sigma, \delta, \widetilde{\mathbf{\Pi}}(\theta), \boldsymbol{\chi}, \mathscr{C}\right), \tag{12}$$

where $\widetilde{\mathbf{\Pi}}(\theta)_{ij}$ is the occurrence probability of event σ_j from state q_i and $\sum_j \widetilde{\mathbf{\Pi}}(\theta)_{ij} = 1 - \theta \ \forall i$. An application of (7) with uniform uniform termination probability θ yields an alternative representation of the sextuple in (12).

$$G(\theta) = \big(Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C}\big), \tag{13}$$

where $\widetilde{\mathbf{P}}$ is the the morph matrix of the underlying Markov chain.

As $\theta \to 0^+$, the resulting non-terminating plant model is denoted as

$$G(0) = (Q, \Sigma, \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C}). \tag{14}$$

Definition 11: Given $\theta \in (0, 1)$, a terminating plant $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$ is defined to be the θ-neighbour of the non-terminating plant $G(0) = (Q, \Sigma, \ \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$.

For a given non-terminating plant $G(0)$ and a fixed $\theta_0 \in (0, 1)$, there is exactly one θ_0-neighbour $G(\theta_0)$.

Notation 2: Let $\theta \in (0, 1)$ be the unform probability of termination for a terminating plant $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$. Let $\mathbf{P}$ be the state transition probability matrix of the underlying Markov chain, which is generated from δ and $\widetilde{\mathbf{\Pi}}$ (see equation (2)). Then, the (renormalized) language measure vector (see Definition 8) is obtained as

$$\boldsymbol{\nu}(\theta) = \theta\Big[\mathbb{I} - (1-\theta)\mathbf{P}\Big]^{-1}\boldsymbol{\chi} \tag{15}$$

where $(1-\theta)\mathbf{P}$ is the sub-stochastic transition probability matrix for the terminating plant. Similarly, for a non-terminating plant $G(0) = (Q, \Sigma, \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$ having the stochastic transition probability matrix $\mathbf{P}$, the (renormalized) measure vector (Chattopadhyay and Ray 2006a) is denoted as

$$\boldsymbol{\nu}(0) = \lim_{\theta \to 0^+} \boldsymbol{\nu}(\theta) = \lim_{\theta \to 0^+} \theta\Big[\mathbb{I} - (1-\theta)\mathbf{P}\Big]^{-1}\boldsymbol{\chi} \tag{16}$$

In the sequel, renormalized measure $\boldsymbol{\nu}$ in equations (10) and (11) is referred to as measure for brevity.

3.2 *Construction of an optimal supervisor*

A supervisor disables a subset of the set $\mathscr{C}$ of controllable transitions and hence there is a bijection between the set of all possible supervision policies and the power set $2^{\mathscr{C}}$. That is, there exists $2^{|\mathscr{C}|}$ possible supervisors and each supervisor is uniquely identifiable with a subset of $\mathscr{C}$ and the language measure ν allows a quantitative comparison of different supervision policies.

Definition 12: For an unsupervised (non-terminating) plant $G(0) = (Q, \Sigma, \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$, let $G^\dagger$ and $G^\ddagger$ be the supervised plants with sets of disabled transitions, $\mathscr{D}^\dagger \subseteq \mathscr{C}$ and $\mathscr{D}^\ddagger \subseteq \mathscr{C}$, respectively, whose measures are $\boldsymbol{\nu}^\dagger$ and $\boldsymbol{\nu}^\ddagger$. Then, the supervisor that disables $\mathscr{D}^\dagger$ is defined to be superior to the supervisor that disables $\mathscr{D}^\ddagger$ if $\boldsymbol{\nu}^\dagger \geqq_{\mathbf{E}} \boldsymbol{\nu}^\ddagger$ and strictly superior if $\boldsymbol{\nu}^\dagger >_{\mathbf{E}} \boldsymbol{\nu}^\ddagger$.

Definition 13 (Optimal supervision problem)**:** Given a (non-terminating) plant $G(0) = (Q, \Sigma, \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$, the problem is to compute a supervisor that disables a subset $\mathscr{D}^\star \subseteq \mathscr{C}$, such that

$$\boldsymbol{\nu}^\star \geqq_{\mathbf{E}} \boldsymbol{\nu}^\dagger \quad \forall \mathscr{D}^\dagger \subseteq \mathscr{C}$$

where $\boldsymbol{\nu}^\star$ and $\boldsymbol{\nu}^\dagger$ are the measure vectors of the supervised plants $G^\star$ and $G^\dagger$ under $\mathscr{D}^\star$ and $\mathscr{D}^\dagger$, respectively.

Remark 2: For a non-trivial plant $G(0) = (Q, \Sigma, \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$ (i.e., $|Q| > 1$), there may exist two supervisors that are not comparable in the sense of Definition 12. For example, given a two-state unsupervised plant G, if $G^\dagger$ and $G^\ddagger$ are supervised plants under two different supervisors with disabled transition sets, $\mathscr{D}^\dagger$ and $\mathscr{D}^\ddagger$, respectively, then the following situation may occur for the indices $i \neq j$.

$$\Big(\nu_i^\dagger > \nu_i^\ddagger\Big) \bigwedge \Big(\nu_j^\dagger < \nu_j^\ddagger\Big),$$

where $\boldsymbol{\nu}_i^\dagger$ and $\boldsymbol{\nu}_i^\ddagger$ are the ith elements of the measure vectors for $G^\dagger$ and $G^\ddagger$, respectively. It is shown in the next section that, for a given plant, an optimal supervisor (in the sense of Definition 13) does exist for which the measure vector is elementwise greater than or equal to the measure vector of the plant under any other supervision policy.

Terminating plant models have sub-stochastic transition probability matrices (see Definition 7). By postulating the existence of unmodelled transitions, such plants can be transformed to non-terminating models as explained below. For uniform termination probability $\theta \in (0, 1)$, equations (8) and (11) suggest the possibility of computing optimal supervision policies for terminating plants based on the analysis of non-terminating plants.

4. Optimal control problem: solution

This section presents a solution to the optimal supervision problem by assuming a uniform non-zero probability of termination, θ, at each state. Then, it is shown that the solution for the corresponding non-terminating plant can be obtained from the control policy of the terminating plant and the bounds on the derivatives of the language measure (see Appendix A).

Let $\theta \in (0, 1)$ be the uniform termination probability of an unsupervised plant $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$. The resulting (substochastic) state transition cost matrix is $\boldsymbol{\Pi}(\theta) = (1-\theta)\mathbf{P}$. For such plants with uniform non-zero termination probability, the following lemma states existence of an augmented plant model.

Lemma 3: For the terminating plant $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$, let the corresponding augmented non-terminating plant be $G_{\text{aug}} = (Q_{\text{aug}}, \Sigma_{\text{aug}}, \delta_{\text{aug}}, \widetilde{\boldsymbol{\Pi}}_{\text{aug}}, \boldsymbol{\chi}_{\text{aug}}, \mathscr{C})$. Let $\boldsymbol{v}^{\star}(\theta)$ and $\boldsymbol{v}^{\dagger}(\theta)$ be the measures of the terminating plant with the respective sets of disabled transitions $\mathscr{D}^{\star} \subseteq \mathscr{C}$ and $\mathscr{D}^{\dagger} \subseteq \mathscr{C}$. Then,

$$\exists \mathscr{D}^{\star} \subseteq \mathscr{C} \quad \text{s.t. } \boldsymbol{v}^{\star}(\theta) \geqq_{\mathbf{E}} \boldsymbol{v}^{\dagger}(\theta) \quad \forall \mathscr{D}^{\dagger} \subseteq \mathscr{C} \quad \forall \theta \in (0,1) \tag{17}$$

which implies that an optimal supervisor for G_{aug} exists (in the sense of Definition 13) which disables $\mathscr{D}^{\star} \subseteq \mathscr{C}$.

Proof: The first n elements of the measure vectors of the augmented plant and the unaugmented plant are identically equal as seen in equation (11). Then, the proof follows from Definition 12. □

The remainder of this section derives an algorithm for a supervision policy that elementwise maximizes the measure of the terminating plant $G(\theta)$. Lemma 3 guarantees that the optimal policy is based on a non-terminating plant.

Proposition 2 (Monotonicity)**:** Let $\boldsymbol{\Pi}(\theta)$ and $\boldsymbol{v}(\theta)$ be the state transition cost matrix and the measure vector of an unsupervised plant $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$, respectively. Let a supervisor be constructed to reconfigure the plant by disabling a set of controllable transitions $\mathscr{D}^{\dagger} \subseteq \mathscr{C}$ such that $\boldsymbol{\Pi}$ is modified to $\boldsymbol{\Pi}^{\dagger}$ by following Algorithm 1. Then, denoting the measure vector of the supervised plant by $\boldsymbol{v}^{\dagger}$, it follows that $\boldsymbol{v}^{\dagger} \geqq_{\mathbf{E}} \boldsymbol{v}$; and equality holds if and only if $\boldsymbol{\Pi}^{\dagger} = \boldsymbol{\Pi}$.

Algorithm 1: Monotonic Increase of Measure $\boldsymbol{v}$

```
begin
    for $i = 1$ to $n$ do
        for $j = 1$ to $n$ do
            $\Pi^{\dagger}_{ij} = \Pi_{ij} + \beta_{ij}$, $\Pi^{\dagger}_{ii} = \Pi_{ii} - \beta_{ij}$   if $\boldsymbol{v}_j > \boldsymbol{v}_i$ with $\beta_{ij} > 0$
            $\Pi^{\dagger}_{ij} = \Pi_{ij}$, $\Pi^{\dagger}_{ii} = \Pi_{ii}$   if $\boldsymbol{v}_j = \boldsymbol{v}_i$
            $\Pi^{\dagger}_{ij} = \Pi_{ij} - \beta_{ij}$, $\Pi^{\dagger}_{ii} = \Pi_{ii} + \beta_{ij}$   if $\boldsymbol{v}_j < \boldsymbol{v}_i$ with $\beta_{ij} > 0$
        endfor
    endfor
end
```

Proof: It follows from equation (15) in Notation 2 that

$$\begin{aligned} \boldsymbol{v}^{\dagger} - \boldsymbol{v} &= \theta\left[\mathbb{I} - \boldsymbol{\Pi}^{\dagger}\right]^{-1} - \theta[\mathbb{I} - \boldsymbol{\Pi}]^{-1}\boldsymbol{\chi} \\ &= \theta\left[\mathbb{I} - \boldsymbol{\Pi}^{\dagger}\right]^{-1}\left([\mathbb{I} - \boldsymbol{\Pi}] - [\mathbb{I} - \boldsymbol{\Pi}^{\dagger}]\right)[\mathbb{I} - \boldsymbol{\Pi}]^{-}1\boldsymbol{\chi} \\ &= \theta\left[\mathbb{I} - \boldsymbol{\Pi}^{\dagger}\right]^{-1}\left(\boldsymbol{\Pi}^{\dagger} - \boldsymbol{\Pi}\right)\boldsymbol{v}. \end{aligned}$$

Defining the matrix $\boldsymbol{\Delta} \triangleq \boldsymbol{\Pi}^{\dagger} - \boldsymbol{\Pi}$, and the ith row of $\boldsymbol{\Delta}$ as Δ_i, it follows that

$$\Delta_i{}^T \boldsymbol{v} = \sum_j \Delta_{ij} v_j = \sum_j \theta \beta_{ij} \Gamma_{ij} \tag{18}$$

where

$$\Gamma_{ij} = \begin{cases} (v_i - v_j) & \text{if } v_i > v_j \\ 0 & \text{if } v_i = v_j \Longrightarrow \Gamma_{ij} \geqq 0 \ \forall i, j. \\ (v_j - v_i) & \text{if } v_i < v_j \end{cases}$$

Since $\sum_{i=1}^{n} \boldsymbol{\Pi}_{ij} = \sum_{i=1}^{n} \boldsymbol{\Pi}^{\dagger}_{ij}$; $\forall j, k$, it follows from non-negativity of $\boldsymbol{\Pi}$, that $[I - \boldsymbol{\Pi}^{\dagger}]^{-1} >_{\mathbf{E}} \mathbf{0}$. Since $\beta_i \geq 0 \ \forall i$, it follows that $\Delta_i{}^T \boldsymbol{v} \geq 0 \ \forall i \Rightarrow \boldsymbol{v}^{\dagger} \geqq_{\mathbf{E}} \boldsymbol{v}$. For $v_j \neq 0$ and Δ as defined above, $\Delta_i^T \boldsymbol{v}^k = 0$ if and only if $\Delta = 0$. Then, $\boldsymbol{\Pi}^{\dagger} = \boldsymbol{\Pi}$ and $\boldsymbol{v}^{\dagger} = \boldsymbol{v}$. □

Corollary 1: Under an identical situation to that assumed in the statement of Proposition 2, let the plant be reconfigured as given in Algorithm 2. Then, denoting the measure vector of the modified plant by $\boldsymbol{v}^{\dagger}$, it follows that $\boldsymbol{v}^{\dagger} \leqq_{\mathbf{E}} \boldsymbol{v}$; and equality holds if and only if $\boldsymbol{\Pi}^{\dagger} = \boldsymbol{\Pi}$.

Algorithm 2: Monotonic Decrease of Measure $\boldsymbol{v}$

```
begin
    for $i = 1$ to $n$ do
        for $j = 1$ to $n$ do
            $\Pi^{\dagger}_{ij} = \Pi_{ij} + \beta_{ij}$, $\Pi^{\dagger}_{ii} = \Pi_{ii} - \beta_{ij}$   if $\boldsymbol{v}_j < \boldsymbol{v}_i$ with $\beta_{ij} > 0$
            $\Pi^{\dagger}_{ij} = \Pi_{ij}$, $\Pi^{\dagger}_{ii} = \Pi_{ii}$   if $\boldsymbol{v}_j = \boldsymbol{v}_i$
            $\Pi^{\dagger}_{ij} = \Pi_{ij} - \beta_{ij}$, $\Pi^{\dagger}_{ii} = \Pi_{ii} + \beta_{ij}$   if $\boldsymbol{v}_j > \boldsymbol{v}_i$ with $\beta_{ij} > 0$
        endfor
    endfor
end
```

Proof: The proof is similar to that of Proposition 26. □

Proposition 2 facilitates formulation of the algorithm for computing an optimal supervisor for plants with uniform non-zero probability of termination at each state. Let the kth iteration of the algorithm compute a set $\mathscr{D}^{[k]} \subseteq \mathscr{C}$ of controllable transitions to be disabled in the sense of the control philosophy in Definition 9. The language measure vector computed in the kth iteration of the algorithm is denoted by $\boldsymbol{\nu}^{[k]}$. The algorithm terminates at the $(k+1)$th iteration if $\mathscr{D}^{[k]} = \mathscr{D}^{[k+1]}$, which is the optimal set of disabled transitions computed by the algorithm and is denoted by $\mathscr{D}^{\star}$. The algorithm is started with the unsupervised plant (i.e., with all controllable transitions enabled) and hence $\mathscr{D}^{[0]} = \emptyset$. A formal description is given in Algorithm 3.

Algorithm 3: Optimal Supervisor Algorithm

```
input  : $\Pi(\theta) = (1-\theta)\mathbf{P}$, $\chi$, $\mathscr{C}$
output: Optimal set of disabled transitions $\mathscr{D}^{\star}$
begin
    Set $\mathscr{D}^{[0]} = \emptyset$ ;            /* Initial disabling set */
    Set $k$ = 1 ;
    Set Terminate = false ;
    while (Terminate == false) do
        Compute $\nu^{[k]}$ ;
        for $j = 1$ to $n$ do
            for $k = 1$ to $n$ do
                Disable all controllable transitions $q_k \xrightarrow{\sigma} q_j$
                such that $\nu_j < \nu_k$ ;
                Enable all controllable transitions $q_k \xrightarrow{\sigma} q_j$
                such that $\nu_j \geqq \nu_k$ ;
            endfor
        endfor
        Compute $\mathscr{D}^{[k]}$ ;            /* $k^{th}$ disabling set */
        if $\mathscr{D}^{[k]} == \mathscr{D}^{[k-1]}$ then
            Terminate = true ;
        else
            $k = k + 1$ ;
        endif
    endw
    $\mathscr{D}^{\star} = \mathscr{D}^{[k]}$ ;            /* Optimal disabling set */
end
```

Proposition 3: Let $\boldsymbol{\nu}^{[k]}$ be the language measure vector computed in the kth iteration of Algorithm 3. The measure vectors computed by the algorithm form an elementwise non-decreasing sequence, i.e., $\boldsymbol{\nu}^{[k+1]} \geqq_{\mathbf{E}} \boldsymbol{\nu}^{[k]}\ \forall k$.

Proof: Let the state transition probability matrix in the kth iteration of Algorithm 3 be denoted by $\boldsymbol{\Pi}^{[k]}$. Then, the matrix $\boldsymbol{\Pi}^{[k+1]}$ is generated from $\boldsymbol{\Pi}^{[k]}$ by following the procedure as described in Proposition 2. Hence, $\boldsymbol{\nu}^{[k+1]} \geqq_{\mathbf{E}} \boldsymbol{\nu}^{[k]}$. □

Proposition 4 (effectiveness)**:** Algorithm 3 is an effective procedure (Hopcroft *et al.* 2001), i.e., it is guaranteed to terminate.

Proof: Let $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$ be the unsupervised plant and let $Card(\mathscr{C}) = \ell \in \mathbb{N}$. Denoting the set of all permutations of the vector $[1\ 2\ \cdots\ \ell]^T$ by $\mathscr{P}(\ell)$, a function $\xi\colon 2^{\mathscr{C}} \longrightarrow \mathscr{P}(\ell)$ is defined as

1. $\forall \mathscr{D}^{\dagger} \subseteq \mathscr{C}, \left(\nu^{\dagger}_{i_1} > \nu^{\dagger}_{i_2} > \cdots > \nu^{\dagger}_{i_n}\right)$
   $\Longrightarrow \left(\xi(\mathscr{D}^{\dagger}) = [i_1\ i_2 \cdots i_n]^T\right)$
2. $\left(\nu^{\dagger}_{i_s} = \nu^{\dagger}_{i_t}\right) \wedge (i_s > i_t)$
   $\Longrightarrow \left(\text{if } \xi(\mathscr{D}^{\dagger})_s = i_s \text{ and } \xi(\mathscr{D}^{\dagger})_t = i_t \text{ then } s > t\right).$

Let $\boldsymbol{\nu}^{[k]}$ be the measure vector computed in the kth iteration of Algorithm 3. Then, $\boldsymbol{\nu}^{[k]} = \boldsymbol{\nu}^{[k+1]}$ implies that Algorithm 3 terminates in $k+1$ iterations according to its stopping rule.

Next let $\mathscr{D}^{[k_1]}$ and $\mathscr{D}^{[k_2]}$ be the disabling sets in iterations k_1 and k_2, respectively. If $\xi(\mathscr{D}^{[k_1]}) = \xi(\mathscr{D}^{[k_2]})$, then $\boldsymbol{\nu}^{[k_1+1]} = \boldsymbol{\nu}^{[k_2+1]}$. Since $\xi(\mathscr{D}^{[k_1]}) = \xi(\mathscr{D}^{[k_2]})$, it follows from the definition of ξ that if $\nu^{[k_1]}_i > \nu^{[k_1]}_j$, then $\nu^{[k_2]}_i \geqq \nu^{[k_2]}_j$. If $\nu^{[k_2]}_i > \nu^{[k_2]}_j$ then controllable transitions $q_i \rightarrow_{\sigma} q_j$ are disabled in both iterations k_1+1 and k_2+1. If $\nu^{[k_1]}_i = \nu^{[k_1]}_j$, then disabling or enabling controllable transitions $q_i \rightarrow_{\sigma} q_j$ does not affect the measure vector. Hence, it follows that $\boldsymbol{\nu}^{[k_1+1]}$ and $\boldsymbol{\nu}^{[k_2+1]}$ can be obtained by disabling the same set of controllable transitions, thus implying $\boldsymbol{\nu}^{[k_1+1]} = \boldsymbol{\nu}^{[k_2+1]}$. Since the measure vectors can repeat only at the final iteration, Algorithm 3 is guaranteed to terminate within $Card(\mathscr{P}(\ell)) = \ell!$ iterations. Therefore, effectiveness of Algorithm 3 is established. □

Next it is established that Algorithm 3 is correct in the sense that an optimal supervision policy is generated.

Proposition 5 (Optimality)**:** For a terminating plant $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$, the supervision policy computed by Algorithm 3 is optimal in the sense of Definition 13.

Proof: Let $G(\theta)$ have the state transition cost matrix $\boldsymbol{\Pi}$, measure $\boldsymbol{\nu}^{[0]}$, no disabled events, i.e., $\mathscr{D}^0 = \emptyset$. Let $G(\theta)$ be configured as the supervised plant $G^{\star}(\theta)$ by application of Algorithm 3 when it stops.

Let $G^{\dagger}$ be another configured plant distinct from $G^{\star}$. Let $\mathscr{D}^{\star} \subseteq \mathscr{C}$ and $\mathscr{D}^{\dagger} \subseteq \mathscr{C}$ be the respective sets of disabled transitions and $\nu^{\star}$ and $\nu^{\dagger}$ be the respective measures for $G^{\star}$ and $G^{\dagger}$; and $\mathscr{D}^{\star} \neq \mathscr{D}^{\dagger}$.

Let the following set differences be denoted as: $\triangle\mathscr{D} \triangleq \mathscr{D}^{\star} \setminus \mathscr{D}^{\dagger}$ and $\nabla\mathscr{D} \triangleq \mathscr{D}^{\dagger} \setminus \mathscr{D}^{\star}$. An application of Algorithm 3 yields

- $\forall i,j\ \nu^{\star}_i > \nu^{\star}_j \Longrightarrow$ all controllable transitions $q_i \underset{\sigma}{\to} q_j$ are disabled.
- $\forall i,j\ \nu^{\star}_i \leqq \nu^{\star}_j \Longrightarrow$ all controllable transitions $q_i \underset{\sigma}{\to} q_j$ are enabled.

To change the plant configuration from $G^\star$ to $G^\dagger$, all transitions in $\triangle\mathscr{D}$ are enabled and all transitions in $\nabla\mathscr{D}$ are disabled. Since any such change requires us to either disable a transition $q_i \to_\sigma q_j$ where $\nu_i^\star \le \nu_j^\star$ or enable a disabled transition $q_i \to_\sigma q_j$ where $\nu_i^\star > \nu_j^\star$, it follows from Corollary 1 that $\boldsymbol{\nu}^\dagger \leqq_{\mathbf{E}} \boldsymbol{\nu}^\star$.

Since $G^\dagger$ is an arbitrary configuration distinct from $G^\star$, it follows that $G^\star$ is an optimal supervision policy in the sense of Definition 13. □

In the reported work on discrete event control of non-probabilistic regular languages (e.g., (Ramadge and Wonham 1987)), the emphasis is on computing the maximally permissive supervisor in the sense that the supervised plant language is the supremal controllable sub-language of the specification. A similar approach is taken for probabilistic regular languages (Garg 1992a, b). In contrast, the measure-theoretic concept in this paper computes a policy that maximizes the elements of the language measure vector elementwise to find a supervisor with maximal performance. Proposition 5 shows that there exists at least one optimal supervisor. Now it is shown that the optimal supervisor computed by Algorithm 3 is unique in the sense of being maximally permissive among all policies that guarantee optimal performance of the supervised plant.

Proposition 6 (uniqueness)**:** Given an unsupervised plant $G(\theta)$, the optimal supervisor $G^\star(\theta)$, computed by Algorithm 3, is unique in the sense that it is maximally permissive among all possible supervision policies with optimal performance. That is, if $\mathscr{D}^\star$ and $\mathscr{D}^\dagger$ are the disabled transition sets, and $\boldsymbol{\nu}^\star$ and $\boldsymbol{\nu}^\dagger$ are the language measure vectors for $G^\star$ and an arbitrarily supervised plant $G^\dagger$, respectively, then

$$\boldsymbol{\nu}^\star \equiv_{\mathbf{E}} \boldsymbol{\nu}^\dagger \Longrightarrow \mathscr{D}^\star \subset \mathscr{D}^\dagger \subseteq \mathscr{C}. \tag{19}$$

Proof: If $G^\star$ and $G^\dagger$ are distinct, then $\mathscr{D}^\# \neq \mathscr{D}^\star$. Given $\boldsymbol{\nu}^\star \equiv_{\mathbf{E}} \boldsymbol{\nu}^\#$, let $G^\star$ be reconfigured to $G^\dagger$ by disabling and/or re-enabling appropriate controllable transitions. It follows from equation (18) that

$$\begin{aligned} 0 = \boldsymbol{\nu}^\dagger - \boldsymbol{\nu}^\star &= \left[\mathbb{I} - \boldsymbol{\Pi}^\dagger\right]^{-1}\left(\boldsymbol{\Pi}^\dagger - \boldsymbol{\Pi}^\star\right)\boldsymbol{\nu}^\star \\ &\Rightarrow \left(\boldsymbol{\Pi}^\dagger - \boldsymbol{\Pi}^\star\right)\boldsymbol{\nu}^\star = 0. \end{aligned} \tag{20}$$

The ith element of $\left(\boldsymbol{\Pi}^\dagger - \boldsymbol{\Pi}^\star\right)\boldsymbol{\nu}^\star$ is expressed as the finite sum of real numbers

$$0 = \left(\left(\boldsymbol{\Pi}^\dagger - \boldsymbol{\Pi}^\star\right)\boldsymbol{\nu}^\star\right)_i = \sum_{r=1}^{\tau} T_r^i, \tag{21}$$

where $0 \le \tau \le 2^{Card(\mathscr{C})}$ and each T_r^i is of the form:

$$T_r^i = \begin{cases} \alpha_r^i(\nu_i^\star - \nu_j^\star) > 0, & \text{if } T_r^i \text{ arises due to disabling} \\ & \qquad q_i \underset{\sigma}{\to} q_j \text{ for some } q_j \in Q \\ \alpha_r^i(\nu_j^\star - \nu_i^\star) \ge 0, & \text{if } T_r^i \text{ arises due to enabling} \\ & \qquad q_i \underset{\sigma}{\to} q_j \text{ for some } q_j \in Q \end{cases} \tag{22}$$

because each α_r^i represents event occurrence probabilities and hence are positive, and the the logic of disabling and re-enabling follows Algorithm 3. Therefore, it follows from equation (22) that $T_r^i = 0 \; \forall r \in \{1, \ldots, \tau\}$.

Hence, it is necessary to re-enable controllable transitions $q_i \to_\sigma q_j$ and disable the self loop at q_i such that $\nu_i^\dagger = \nu_j^\dagger$ for reconfiguration from $G_\theta^\dagger$ to $G_\theta^\star$. Note that all such transitions are guaranteed to be enabled in $G_\theta^\star$ (see line 10 in Algorithm 3). Therefore, given $\boldsymbol{\nu}^\star \equiv_{\mathbf{E}} \boldsymbol{\nu}^\dagger$, it follows that $\mathscr{D}^\star \subseteq \mathscr{D}^\dagger$. That is, $G^\star(\theta)$ is unique for all $\theta \in (0,1)$ in the sense that the configured plant is maximally permissive among all other configurations that yield the same optimal measure $\boldsymbol{\nu}^\star(\theta)$. □

4.1 *Optimal control of non-terminating plants*

This section presents the optimal supervision problem for non-terminating plants (i.e., with termination probability $\theta = 0$ at each state) having the structure $G(0) = (Q, \Sigma, \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$ and the corresponding stochastic transition probability matrix is $\mathbf{P}$. The rationale for working on a terminating plant, instead of the non-terminating plant is explained below.

By maximizing the measure $\boldsymbol{\nu}(\theta)$ for a given $\theta \in (0,1)$, an optimal control law can be derived based on the state transition cost matrix $\boldsymbol{\Pi}(\theta) = (1-\theta)\mathbf{P}$ of the supervised plant language and the originally assigned $\boldsymbol{\chi}$-vector. Such an optimal control law is sought to be θ-independent in the sense of having the same disabling set $\mathscr{D} \subseteq \mathscr{C}$ for a given range of θ, where θ might be restricted to be not too far away from 0^+. On the other hand, from the perspective of numerical stability and accuracy in computation of $\boldsymbol{\nu}(\theta)$ (see Definition 8), it is desirable to have a relatively large positive value of θ. The results derived in this section serve toward establishing upper bounds on θ for which the optimal control law should be θ-independent and the associated computation is numerically stable. The main objective is summarized below.

> A uniform non-zero probability of termination $\theta_\star \in (0,1)$ is to be computed such that the terminating plant $G(\theta_\star)$ and the

non-terminating plant $G(0)$ shall have the same the disabling set $\mathscr{D} \subseteq \mathscr{C}$. However, in general, their measures could be different, i.e., $\boldsymbol{\nu}(\theta_\star) \neq_{\mathbf{E}} \boldsymbol{\nu}(0)$.

Proposition 7: Let $(1-\theta)\mathbf{P}$ and $\boldsymbol{\nu}(\theta)$ be the state transition cost matrix and the measure of the plant $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$. Then, for all $q_i, q_j \in Q$, there exists $\theta^\star_{ij} \in (0, 1]$ such that $\forall \theta \in (0, \theta^\star_{ij})$, the sign of $\big(\nu_i(\theta) - \nu_j(\theta)\big)$ is fixed (i.e., positive, negative or zero); and $\theta^\star_{ij}$ can be computed as an explicit function of the stochastic matrix $\mathbf{P}$ and state characteristic vector $\boldsymbol{\chi}$.

Proof: Let $\gamma_{ij}(\theta) \triangleq \nu_i(\theta) - \nu_j(\theta)$ $\forall \theta \in (0, 1)$, which is a smooth function of θ, and $\gamma_{ij}(0) = \lim_{\theta \to 0^+} \gamma_{ij}(\theta)$. The proof is based on the following two cases.

Case 1: No sign change of $\gamma_{ij}(\theta)$ in $(0, 1)$ $\Rightarrow$ $\theta^\star_{ij} = 1$. This includes: $\gamma_{ij}(0) = 0$ and $(d^k \gamma_{ij}(\theta)/d\theta^k)|_{\theta=0} = 0$ for all $k \geq 0$ because $\gamma_{ij}(\theta) = 0$ $\forall \theta \in (0, 1)$ by Proposition A.3.

Case 2: $\gamma_{ij}(\theta)$ changes sign in $(0, 1)$; and $(\partial^r \gamma_{ij}(\theta)/\partial\theta^r)|_{\theta=0} = \lambda_{ij} \neq 0$ for some integer $r \geq 0$.

If $r = 0$, there exists $\tau_1 \in (0, 1)$ such that $\gamma_{ij}(\tau_1) = 0$ for the first time. If $r > 0$, it is possible that $\gamma_{ij}(0) = 0$. Then, as θ is increased from zero, $\gamma_{ij}(\theta)$ becomes non-zero and there exists $\tau_1 \in (0, 1)$ such that $\gamma_{ij}(\tau_1) = 0$ again. Smoothness of $\gamma_{ij}(\theta)$ necessitates that $(\partial^r \gamma_{ij}(\theta)/\partial\theta^r)|_{\theta=\theta^\star_{ij}} = 0$ for some $\theta^\star_{ij} \in (0, \tau_1)$. Then, it follows from the Mean value Theorem that there exists $\tau_2 \in (0, \theta^\star_{ij})$ such that

$$\frac{\partial^{r+1} \gamma_{ij}(\theta)}{\partial \theta^{r+1}}\Big|_{\theta=\tau_2} = \frac{\lambda_{ij}}{\theta^\star_{ij}}$$

for the given $r \geq 0$, Proposition A.2, triangular inequality, and the relation $\gamma_{ij}(\theta) = \nu_i(\theta) - \nu_j(\theta)$ yield

$$\theta^\star_{ij} = \frac{\left| (\partial^r \nu_i(\theta)/\partial\theta^r)\big|_{\theta=0} - (\partial^r \nu_j(\theta)/\partial\theta^{r+1})\big|_{\theta=0} \right|}{(r+1)!\; 2^{r+3} \left(inf_{\alpha \neq 0} \left\| \left[\mathbb{I} - P + \alpha\mathscr{P}\right]^{-1} \right\|_\infty \right)^{r+1}}$$

$$= \begin{cases} \dfrac{|\{[\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1}\chi - \mathscr{P}\chi\}_i - \{[\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1}\chi - \mathscr{P}\chi\}_j|}{8 \times \left(inf_{\alpha \neq 0} \|[\mathbb{I} - \mathbf{P} + \alpha\mathscr{P}]^{-1}\|_\infty\right)}, & \text{if } r = 0 \\[3ex] \dfrac{\left| \begin{matrix} \left\{[\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1}\left[\mathbb{I} - [\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1}\right]^r \chi\right\}_i \\ -\left\{[\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1}\left[\mathbb{I} - [\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1}\right]^r \chi\right\}_j \end{matrix} \right|}{2^{r-3}\left(\inf_{\alpha \neq 0} \left\| \left[\mathbb{I} - P + \alpha\mathscr{P}\right]^{-1} \right\|_\infty\right)^{r+1}}, & \text{if } r > 0. \end{cases} \tag{23}$$

□

Remark 3: For a non-terminating plant $G(0) = (Q, \Sigma, \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi})$, let $\theta_\star = \min_{i,j} \theta^\star_{ij}$. Then, the plant configuration obtained by applying a single iteration of Algorithm 3 to the θ-parameterized plant $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$ is identical for all $\theta \leq \theta_\star$.

The procedure of computing $\theta_\star$ is summarized as Algorithm 4.

Proposition 8: Complexity of computing a positive bound for $\theta_\star$ is $O(n^3)$ where n is the number of plant states.

Algorithm 4: Computation of the Bound $\theta_\star$

```
input  : P, $\chi$
output: $\theta_\star$
begin
    Set $\theta_\star = 1$;
    Set $\theta_{curr} = 0$;
    Compute $\mathscr{P}$ ;    /* See Algorithm 6 in Appendix A */
    Compute $M_0 = \left[\mathbb{I} - \mathbf{P} + \mathscr{P}\right]^{-1}$ ;
    Compute $M_1 = \left[\mathbb{I} - \left[\mathbb{I} - \mathbf{P} + \mathscr{P}\right]^{-1}\right]$;
    Compute $M_2 = inf_{\alpha \neq 0} \left\|\left[\mathbb{I} - \mathbf{P} + \alpha\mathscr{P}\right]^{-1}\right\|_\infty$;
    for $j = 1$ to $n$ do
        for $i = 1$ to $n$ do
            if $(\mathscr{P}\chi)_i - (\mathscr{P}\chi)_j \neq 0$ then
                $\theta_{curr} = \frac{1}{8M_2}\left|(\mathscr{P}\chi)_i - (\mathscr{P}\chi)_j\right|$
            else
                for $r = 0$ to $n$ do
                    if $(M_0\chi)_i \neq (M_0\chi)_j$ then
                        Break;
                    else
                        if $\left(M_0 M_1^r \chi\right)_i \neq \left(M_0 M_1^r \chi\right)_j$ then
                            Break;
                        endif
                    endif
                endfor
                if $r == 0$ then
                    $\theta_{curr} = \frac{|(M_0 - \mathscr{P})\chi_i - (M_0 - \mathscr{P})\chi_j|}{8M_2}$;
                else
                    if $r > 0$ AND $r \leq n$ then
                        $\theta_{curr} = \frac{|(M_0 M_1 \chi)_i - (M_0 M_1 \chi)_j|}{2^{r+3} M_2}$ ;
                    else
                        $\theta_{curr} = 1$ ;
                    endif
                endif
            endif
            $\theta_\star = \min(\theta_\star, \theta_{curr})$ ;
        endfor
    endfor
end
```

Proof: Referring to Algorithm 4, the part within the nested For loops (lines 10 to 32) is executed at most n^2 times and each iteration involves only single-iteration scalar operations. Thus the computational complexity of this part is of the order of $O(n^2)$. Lines 5 and 6 involve inversion of $n \times n$ dimensional non-singular matrices and hence the complexity of execution is of the order of $O(n^3)$. Proposition G (see Appendix) guarantees that the complexity of computing $\mathscr{P}$ is, in general, of the order of $O(n^3)$. Line 7, which computes $M_2 = \inf_{\alpha \neq 0} \|[\mathbb{I} - \mathbf{P} + \alpha_0 \mathscr{P}]^{-1}\|_\infty$, is a search problem. However, since M_2 appears only in the denominator of the expressions for θ_{curr}, it follows that, if for some $\alpha = \alpha_0 \neq 0$ and by using

$$M_2 = \left\| \left[\mathbb{I} - \mathbf{P} + \alpha_0 \mathscr{P}\right]^{-1} \right\|_\infty \tag{24}$$

it is possible to obtain a positive lower bound of $\theta_\star$ in Algorithm 4. Since the computation of $\|[\mathbb{I} - \mathbf{P} + \alpha_0 \mathscr{P}]^{-1}\|_\infty$ is of the order of $O(n^3)$ due to the matrix inversion, it is concluded that a positive lower bound of $\theta_\star$ can be computed with a complexity of $O(n^3)$. □

Remark 4: It is shown in (Chattopadhyay and Ray 2006a) that for any stochastic matrix $\mathbf{P}$

$$\begin{aligned}\left[\mathbb{I}-\mathbf{P}+\alpha\mathscr{P}\right]^{-1} &= \left[\mathbb{I}-\mathbf{P}+\mathscr{P}\right]^{-1} \\ &\quad + \left(\frac{1-\alpha}{\alpha}\right)\mathscr{P} \;\; \forall \alpha \neq 0 \\ &\Longrightarrow \left(\left[\mathbb{I}-\mathbf{P}+\mathscr{P}\right]^{-1}-\mathscr{P}\right)+\left(\frac{1}{\alpha}\right)\mathscr{P}. \end{aligned} \tag{25}$$

Using $M_2 = \|[\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1}\|_\infty$ instead of $M_2 = \inf_{\alpha \neq 0} \|[\mathbb{I} - \mathbf{P} + \alpha \mathscr{P}]^{-1}\|_\infty$ (i.e., using $\alpha = 1$) in Algorithm 4 yields a value which satisfies the requirement stated in Remark 3 and therefore qualifies as $\theta_\star$. Thus, the major advantage of this approximation is having significantly smaller computational complexity because the search involved in computing the infimum is avoided at the cost of using a smaller value of $\theta_\star$, which may make subsequent computation of measure slightly more difficult due to possible ill-conditioning (see Definition 8).

On account of Proposition 7 and Remark 3, Algorithm 3 is modified to solve the optimal supervision problem for non-terminating plants and the modified version is formally presented in Algorithm 5.

Algorithm 5: Computation of Optimal Supervisor

```
input  : $\mathbf{P}$, $\chi$, $\mathscr{C}$
output: Optimal set of disabled transitions $\mathscr{D}^\star$
begin
    Set $\mathscr{D}^{[0]} = \emptyset$ ;            /* Initial disabling set */
    Set $\widetilde{\Pi}^{[0]} = \widetilde{\Pi}$ ;    /* Initial event prob. matrix */
    Set $\theta_\star^{[0]} = 0.99$;
    Set $k$ = 1 ;
    Set Terminate = false ;
    while (Terminate == false) do
        Compute $\theta_\star^{[k]}$;                    /* Algorithm 4 */
        Set $\widetilde{\Pi}^{[k]} = \frac{1-\theta_\star^{[k]}}{1-\theta_\star^{[k-1]}} \widetilde{\Pi}^{[k-1]}$;
        Compute $\boldsymbol{\nu}^{[k]}$ ;
        for $j = 1$ to $n$ do
            for $i = 1$ to $n$ do
                Disable all controllable transitions $q_i \xrightarrow{\sigma} q_j$
                such that $\boldsymbol{\nu}_j^{[k]} < \boldsymbol{\nu}_i^{[k]}$ ;
                Enable all controllable transitions $q_i \xrightarrow{\sigma} q_j$
                such that $\boldsymbol{\nu}_j^{[k]} \geqq \boldsymbol{\nu}_i^{[k]}$ ;
            endfor
        endfor
        Collect all disabled transitions in $\mathscr{D}^{[k]}$;
        if $\mathscr{D}^{[k]} == \mathscr{D}^{[k-1]}$ then
            Terminate = true ;
        else
            $k$ = $k$ + 1 ;
        endif
    endw
    $\mathscr{D}^\star$ = $\mathscr{D}^{[k]}$ ;          /* Optimal disabling set */
end
```

Proposition 9 (effectiveness)**:** Algorithm 5 is an effective procedure (Hopcroft *et al.* 2001), i.e., it is guaranteed to terminate.

Proof: Comparison of Algorithm 3 and Algorithm 5 reveals that while the former assumes a fixed probability of termination θ at each state, the latter modifies this parameter, denoted as $\theta_\star^{[k]}$, at each iteration k. Let $\theta_{\min} = \min\left(\theta_\star^{[1]}, \theta_\star^{[2]}\right)$ and let $\mathscr{D}^{[1]}(\theta_{\min})$ and $\mathscr{D}^{[2]}(\theta_{\min})$ be sets of disabled transition at the first and second iterations, respectively, for the terminating plant $G(\theta_{\min})$. Similarly, for the non-terminating plant $G(0)$, let $\mathscr{D}^{[1]}(0)$ and $\mathscr{D}^{[2]}(0)$ be the sets of disabled transitions at the first and second iterations, respectively. It follows from Remark 3 that $\mathscr{D}^{[1]}(0) = \mathscr{D}^{[1]}(\theta_{\min})$ and $\mathscr{D}^{[2]}(0) = \mathscr{D}^{[2]}(\theta_{\min})$.

Extending the above argument by induction based on k iterations of Algorithm 5 and denoting $\theta_{\min} = \min(\theta_\star^{[1]}, \ldots, \theta_\star^{[k]})$, an application of Algorithm 3 on a terminating plant $G(\theta_{\min})$ yields

$$\mathscr{D}^{[r]}(0) = \mathscr{D}^{[r]}(\theta_{\min}) \; \forall r \in \{1, \ldots, k\}.$$

Proposition 4 states that, for an arbitrary plant, Algorithm 3 is guaranteed to terminate within finitely many iterations. Hence, Algorithm 5 is an effective procedure. □

Next, it is shown that the plant configuration obtained by Algorithm 5 is optimal in the sense of Definition 13.

Proposition 10 (optimality)**:** For a non-terminating plant $G(0) = (Q, \Sigma, \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$), the supervision policy computed by Algorithm 5 is optimal in the sense of Definition 13.

Proof: Let the set of disabled transitions computed at the kth iteration Algorithm 5 be denoted by $\mathscr{D}^{[k]}_{\lim}$ and the termination probability be denoted by $\theta^{[k]}_{\star}$. Let the set of disabled transitions at the convergence of Algorithm 5 be $\mathscr{D}^{[m]}_{\lim}$. Let $\theta_{\min} = \min_{r\in\{1,\ldots,\ell\}} (\theta^{[1]}_{\star}, \ldots, \theta^{[\ell]}_{\star}) > 0$.

Let $G(\theta_{\min})$ be a terminating plant with $\boldsymbol{\Pi}(\theta_{\min}) = (1-\theta_{\min})\mathbf{P}$. It follows from the proof of Proposition 9 that applications of Algorithm 3 to $G(\theta_{\min})$ and Algorithm 5 to $G(0)$ yield the same set $\mathscr{D}$ of disabled controllable events although the optimal measures, being θ-dependent would be different, i.e., $\boldsymbol{v}(\theta_{\min}) \neq_{\mathbf{E}} \boldsymbol{v}(0)$.

Proposition 5 implies that the optimal disabling set for a plant $G(\theta)$ generates the the same set of disabled controllable transitions for all $0 < \theta \leqq \theta_{\min}$. Because of continuity of $\boldsymbol{v}(\theta)$ with respect to θ, it is argued that $G^{\star}(0)$ is optimal in the sense of Definition 13, i.e., $\boldsymbol{v}^{\star} \geqq_{\mathbf{E}} \boldsymbol{v}^{\dagger}$, where $G^{\dagger}(0)$ is obtained by arbitrarily disabling controllable transitions in G. This completes the proof. □

Next it is shown that the supervision policy computed by Algorithm 5 is unique in the same sense as Proposition 6.

Proposition 11 (Uniqueness)**:** Let $G(0)$ be an unsupervised non-terminating plant and $G^{\star}(0)$ be the supervised plant configured by Algorithm 5. Then, $G^{\star}$ is unique in the sense that it is maximally permissive among supervised plants that yield optimal performance based on θ-neighbours $G(\theta)$ of $G(0)$ (see Definition 11) for all $\theta \in (0, \theta_{\star})$, where $\theta_{\star}$ is computed by Algorithm 4. Equivalently, if $G^{\dagger}(\theta)$ is an arbitrarily supervised plant, then the following condition holds:

$$\left(\boldsymbol{v}^{\star}(\theta) \geqq_{\mathbf{E}} \boldsymbol{v}^{\dagger}(\theta)\right) \bigwedge \left(\left(\mathscr{D}^{\star} \subseteq \mathscr{D}^{\dagger}\right) \bigvee \left(\boldsymbol{v}^{\star}(\theta) \neq_{\mathbf{E}} \boldsymbol{v}^{\dagger}(\theta)\right) \right),$$

where $\boldsymbol{v}$ and $\mathscr{D}$ denote respective language measures and sets of disabled transitions.

Proof: It follows from Proposition 10 that $\boldsymbol{v}^{\star}(0) \geqq_{\mathbf{E}} \boldsymbol{v}^{\dagger}(0)$. It also follows from Proposition that $\boldsymbol{v}^{\star}(\theta) \geqq_{\mathbf{E}} \boldsymbol{v}^{\dagger}(\theta)$ for $\theta \in (0, \theta_{\star})$. If $\boldsymbol{v}^{\star}(\theta) \equiv_{\mathbf{E}} \boldsymbol{v}^{\dagger}(\theta)$, then $G^{\star}(\theta)$ and $G^{\dagger}(\theta)$ are both optimal supervised configurations of the unsupervised terminating plant $G(\theta)$. It follows from Proposition 6 that $\mathscr{D}^{\star} \subseteq \mathscr{D}^{\dagger}$; otherwise $\boldsymbol{v}^{\star}(\theta) \neq_{\mathbf{E}} \boldsymbol{v}^{\dagger}(\theta)$. □

Proposition 12: Computational complexity of Algorithm 5 is of the same order as that of Algorithm 3.

Proof: Algorithm 5 computes $\theta_{\star}$ in each iteration and complexity of this computation is $O(n^3)$, where n is the number of states in the plant (see Proposition 8). Each iteration of both Algorithm 3 and Algorithm 5 involves computation of the measure vector $\boldsymbol{v}$, whose complexity is also $O(n^3)$ because of $n \times n$ matrix inversion. Hence, computational complexity of each iteration is $O(n^3)$ for both Algorithm 3 and Algorithm 5. Finally, the argument presented in Proposition 9 implies that the number of iterations in Algorithm 5 is of the same order as that in Algorithm 3. This completes the proof. □

4.2 *Testing of computational complexity*

Proposition 4 shows that Algorithm 3 is an effective procedure (Hopcroft *et al.* 2001), i.e., the solution is guaranteed to converge in a finite number of iterations. Extensive simulation suggests that the the maximum number of iterations for Algorithm 3 is actually of polynomial order in n, where n is the number of states in the unsupervised plant. The result is illustrated in figure 6, where the maximum number of required iterations $I_{\max}$ is plotted against number, n, of plant states. For each n, 10,000 simulation runs were conducted for synthesis of optimal plant configuration with randomly generated entries in the pair $\left((1-\theta)\mathbf{P},\ \boldsymbol{\chi}\right)$; and $I_{\max}$ was chosen to be the maximum number of iterations required by Algorithm 3 to converge; this is the most conservative case. The plot in figure 1 shows a distinct sub-linear variation. The following conjecture is made based on these observations.

Conjecture 1 (polynomial convergence)**:** *Given a terminating plant $G(\theta)$ with a uniform non-zero probability of termination θ at each of the n plant states,*

1. *Algorithm* 3 *converges in at most $O(n)$ iterations.*
2. *Computational complexity of Algorithm* 3 *is bounded by $O(n^4)$.*

Statement 2 in Conjecture 1 follows from Statement 1 and the following facts: Each iteration has complexity of $O(n^3)$ due to matrix inversion in the computation of

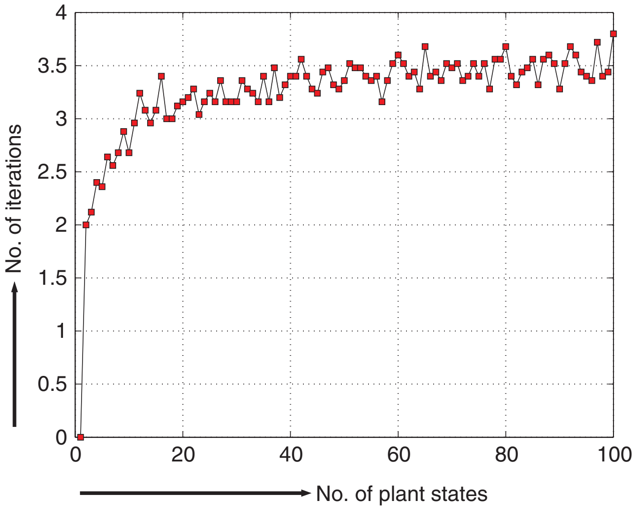


Figure 1. Number of iterations to converge in Algorithm 3.

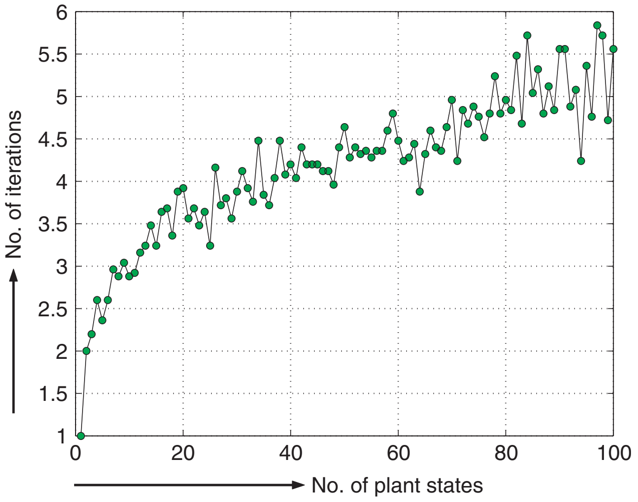


Figure 2. Number of iterations to converge in Algorithm 5.

the language measure vector, and matrix inversion has complexity of $O(n^3)$). Combination of Conjecture 1 and Proposition 12 implies that Algorithm 5 converges in $O(n)$ iterations and that complexity of the algorithm is $O(n^4)$. Similar to the procedure, described above for Algorithm 3, 10, 000 random simulation runs for each n were conducted for testing Algorithm 5. Figure 2 shows the plot of average number of iterations required to converge at each value of n in contrast to figure 1, where the maximum number of iterations is potted. As expected, the plot of figure 2 is also sub-linear.

5. Optimal control of three processor message decoding

This section presents the design of a discrete-event (controllable) supervisor for a multiprocessor message decoding system, described in an earlier publication (Ray *et al*. 2004). The optimal supervisory algorithm has been synthesized based on the algorithms presented in earlier sections.

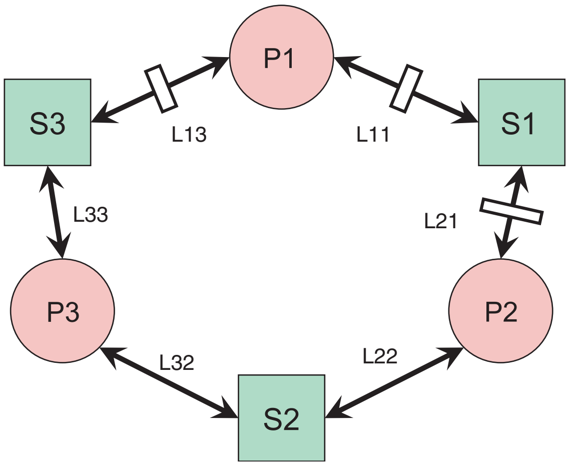


Figure 3. Arrangement of the processor links.

Figure 3 depicts the arrangement of the message decoding system, where each of the three processors, p_1, p_2 and p_3, receives encoded messages that are to be decoded. The processor p_3 normally receives the most important messages, and p_1 receives the least important messages. There is a server between each pair of processors—s_1 between p_1 and p_2; s_2 between p_2 and p_3; and s_3 between p_3 and p_1. Each server is connected to each of its two adjacent processors by a link—the server s_j is connected to the adjacent processors p_i and p_k through the links L_{ij} and L_{kj}, respectively. Out of these six links, each of the three links, L_{11}, L_{12}, and L_{21}, is equipped with a switch to disable the respective connection whenever it is necessary; each of the remaining three links, L_{22}, L_{32}, and L_{33}, always remain connected. Each server s_i is equipped with a decoding key k_i that, at any given time, can only be accessed by only one of the two processors, adjacent to the server, through the link connecting the processor and the server. In order to decode the message, the processor holds the information on both keys of the servers next to it, one at a time. After decoding, the processor simultaneously releases both keys so that other processors may obtain access to them.

Figure 4 depicts the unsupervised plant model of the decoding system as a finite state automaton, where state 1 is the initial state. The event p_{ij} indicates that processor p_i has accessed the key k_j; and the event f_i indicates that the processor p_i has finished decoding and (simultaneously) released both keys in its possession upon completion of decoding. The events f_i are uncontrollable because, after the decoding is

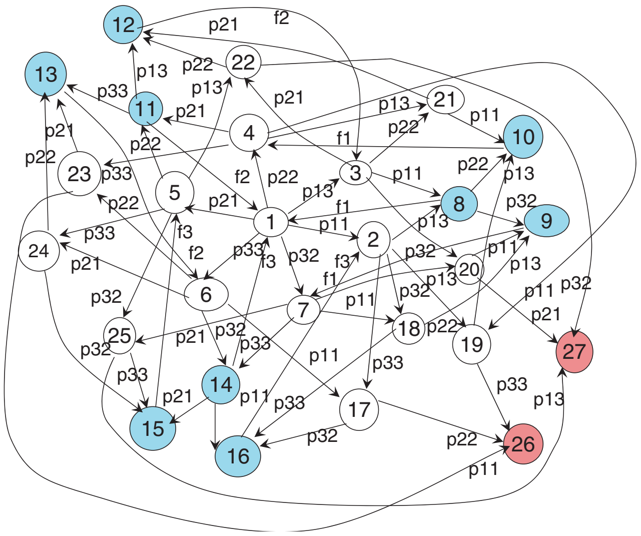


Figure 4. Finite state model of the message decoding system.

initiated, there is no control on when a processor finishes decoding.

Table 1 lists the event cost matrix $\widetilde{\Pi}$. Two different control specifications are investigated. The first set of specifications, which emphasizes avoiding deadlock, is represented by the χ vector in the first column of table 2. The second set of specifications, which focuses on increasing the throughput of processor 1, is represented by the χ vector in the second column of table 2. The positive elements of the χ vector are assigned to the states 8 to 16 that represent successful decoding of each processor. The χ values of the deadlock states 26 and 27, where each processor holds exactly one key and hence no processor releases its key, are assigned negative values. The remaining states are non-marked and are assigned zero weights.

Algorithm 5 is applied to obtain the sequence of measure vectors for the two control specifications. The results of successive iterations, enumerating the renormalized measure vectors, are presented in Table 3 and 4 respectively. The last column in each table is the optimal renormalized measure vector. The optimization requires 7 iterations in Case 1 and 5 iterations in Case 2.

The optimal configurations for the plant obtained under Algorithm 5 are depicted in figures 5 and 6 respectively. For supervisor policy 1, the controlled plant is not trim and, for supervisor policy 2, there are disconnected states in the controlled model. This is interpreted as the supervisor successfully preventing the plant from visiting these states. The critical values for the termination probability $\theta_\star$ computed by the optimization algorithm for each control specification is shown in figure 7.

Table 1. Event occurrence probabilities for processor models.

$p11$	$p13$	$p21$	$p22$	$p32$	$p33$	$f1$	$f2$	$f3$
0.16	0.04	0.16	0.16	0.16	0.32	0.00	0.00	0.00
0.00	0.16	0.00	0.26	0.26	0.32	0.00	0.00	0.00
0.37	0.00	0.21	0.21	0.21	0.00	0.00	0.00	0.00
0.32	0.11	0.26	0.00	0.00	0.32	0.00	0.00	0.00
0.00	0.11	0.00	0.28	0.28	0.33	0.00	0.00	0.00
0.25	0.00	0.25	0.25	0.25	0.00	0.00	0.00	0.00
0.28	0.11	0.28	0.00	0.00	0.33	0.00	0.00	0.00
0.00	0.00	0.00	0.39	0.39	0.00	0.22	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.14	0.00	0.00	0.00	0.43	0.00	0.43	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
0.33	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.34
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
0.00	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.25	0.00	0.00	0.00	0.75	0.00	0.00	0.00
0.50	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00
0.50	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00
0.50	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00
0.00	0.25	0.00	0.00	0.00	0.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.33	0.34
0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.33	0.34

Table 2. Vectors for control specifications.

Case 1			Case 2		
0.000	0.010	0.000	0.000	1.000	0.000
0.000	0.020	0.000	0.000	0.020	0.000
0.000	0.020	0.000	0.000	0.020	0.000
0.000	0.020	0.000	0.000	0.020	0.000
0.000	0.040	0.000	0.000	0.040	0.000
0.000	0.040	0.000	0.000	0.040	0.000
0.000	0.040	0.000	0.000	0.040	0.000
0.010	0.000	−1.000	1.000	0.000	−0.200
0.010	0.000	−1.000	1.000	0.000	−0.200

Next the stable probability distributions of the plant states are compared for the following three cases:

- Open-loop or unsupervised plant
- Plant with the optimal supervision policy for specification 1
- Plant with the optimal supervision policy for specification 2

The distributions are obtained by considering the first row of the matrix $\mathscr{P}$, based on the measure ν_1

Table 3. Iteration vectors for multi-processor model: case 1.

Itr 1	Itr 2	Itr 3	Itr 4	Itr 5	Itr 6	Itr 7
−0.0616	0.0006	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0001	0.0063	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0003	0.0055	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0007	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0011	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0012	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0000	0.0093	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0002	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0000	0.0093	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0007	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0007	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0003	0.0055	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0013	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0010	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0011	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0001	0.0063	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0001	0.0000	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0000	0.0093	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0009	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0000	0.0000	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0003	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0003	0.0000	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0014	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0012	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0012	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0007	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0007	0.0110	0.0124	0.0124	0.0143	0.0143

Table 4. Iteration vectors for multi-processor model: case 2.

Itr 1	Itr 2	Itr 3	Itr 4	Itr 5
0.0598	0.2076	0.2879	0.3245	0.3245
0.0598	0.2074	0.2880	0.3245	0.3245
0.0598	0.2101	0.2879	0.3245	0.3245
0.0598	0.2167	0.2876	0.3232	0.3245
0.0598	0.2109	0.2878	0.3236	0.3245
0.0598	0.2084	0.2882	0.3245	0.3245
0.0598	0.2059	0.2878	0.3245	0.3245
0.0598	0.2090	0.2879	0.3245	0.3245
0.0598	0.2059	0.2878	0.3245	0.3245
0.0598	0.2175	0.2875	0.3230	0.3245
0.0598	0.2089	0.2879	0.3245	0.3245
0.0598	0.2105	0.2879	0.3245	0.3245
0.0598	0.2086	0.2882	0.3245	0.3245
0.0598	0.2078	0.2879	0.3245	0.3245
0.0598	0.2114	0.2878	0.3235	0.3245
0.0598	0.2076	0.2880	0.3245	0.3245
0.0598	0.2080	0.2880	0.3245	0.3245
0.0598	0.2059	0.2878	0.3245	0.3245
0.0598	0.2216	0.2872	0.3241	0.3245
0.0598	0.2059	0.2878	0.3245	0.3245
0.0598	0.2147	0.2879	0.3245	0.3245
0.0598	0.2116	0.2879	0.3245	0.3245
0.0598	0.2084	0.2879	0.3245	0.3245
0.0598	0.2105	0.2878	0.3232	0.3245
0.0598	0.2110	0.2878	0.3236	0.3245
0.0598	0.2077	0.2879	0.3245	0.3245
0.0598	0.2077	0.2879	0.3245	0.3245

corresponding to state 1 which is the initial state in both cases. If the stochastic matrix **P** is primitive (i.e., irreducible and acyclic), then all rows of $\mathscr{P}$ would be identical. However, primitiveness of **P** is not guaranteed even if the unsupervised plant model have this property because any subsequent event disabling may cause loss of reducibility or acyclic properties.

The results on evolution of the distribution are plotted in figure 8. While the unsupervised plant has a finite probability of reaching the deadlock states 26 and 27, the optimal supervisors in both cases successfully prevent occurrence of deadlock in the sense that the stable occupation probabilities for states 26 and 27 are zero for each supervisor. However, supervisor 2 increases the throughput of processor 1 as seen from the increased probability of occupying states 1 and 2.

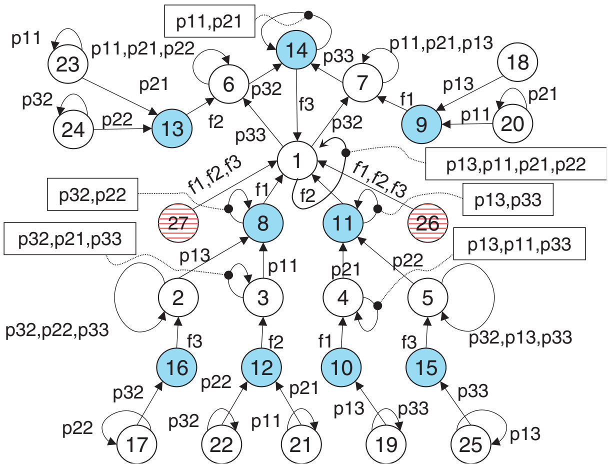


Figure 5. Optimal plant configuration for specification 1.

6. Summary, conclusions, and recommendations for future work

This paper presents the theory, formulation, and validation of optimal supervisory control policies for dynamical systems, modelled as probabilistic finite state automata. The procedure for synthesis of the optimal control policy relies on a (renormalized) signed real measure of regular languages (Chattopadhyay and Ray 2006a) to construct the performance index. The language measure is based on the state transition

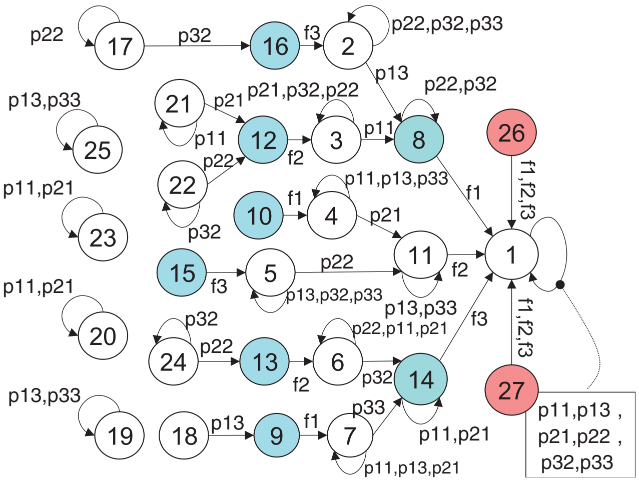


Figure 6. Optimal plant configuration for specification 2

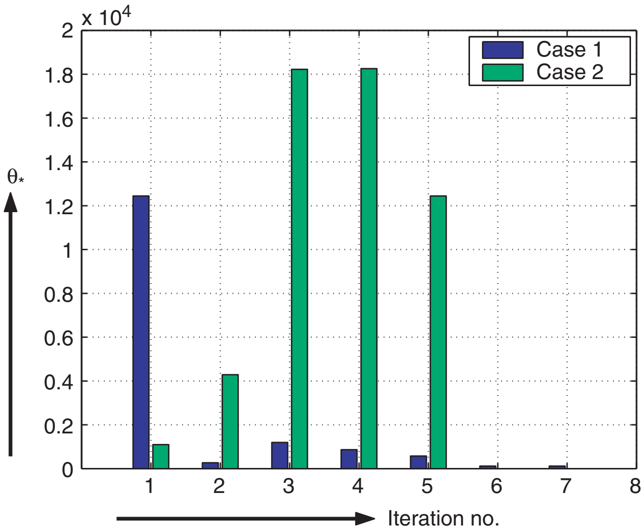


Figure 7. Computed $\theta_\star$ for each iteration.

probability matrix of the underlying finite-state Markov chain model of the process and a characteristic vector of state weights, which serves as the control specification.

The main contribution of the paper is reformulation of the optimal supervisor synthesis algorithm (Ray *et al*. 2004, 2005) for probabilistic finite state plant models in terms of the renormalized measure and extension of the technique to general non-terminating probabilistic models. Specifically, the work reported in this paper removes a fundamental restriction of earlier analysis (Ray *et al*. 2004, Ray 2005), namely, each row sum of the state transition cost matrix $\mathbf{\Pi}$ being strictly less than one, instead of being exactly equal to one. The novel concept of language-based control synthesis, presented in this paper, allows quantification of plant performance instead of solely relying on its qualitative performance (e.g., permissiveness), which is the current state of the art for discrete event supervisory control (Ramadge and Wonham 1987, Cassandras and Lafortune 1999).

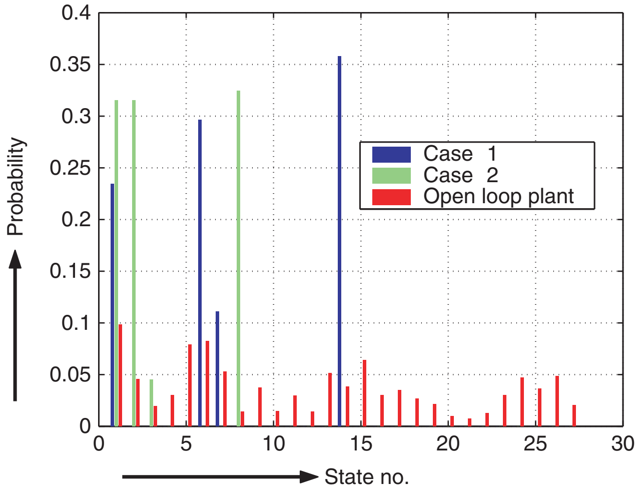


Figure 8. Stable state probability: unsupervised and supervised plants

The following conclusion is drawn in view of using the language measure for construction of the performance index for deriving an optimal control policy. Like any other optimization procedure, it is possible to choose different performance indices to arrive at different optimal policies for discrete event supervisory control. Nevertheless, usage of the language measure provides a systematic procedure for precise comparative evaluation of different supervisors so that the optimal control policy(ies) can be unambiguously identified. These theoretical results also lay the foundation for extension of the language-measure-theoretic framework to plant modelling and control, where all events may not be observable at the supervisory level.

The paper provides details of the algorithms that are required for synthesis of the optimal supervisory control policy. These algorithms are executable in real time on commercially available platforms. Computational complexity of the presented algorithms is polynomial in the number of plant model states. The concepts are elucidated with simple examples and a relevant engineering example. As such it is straight-forward to develop real-time software codes in standard languages, based on the algorithms provided in this paper.

There are several issues that need to be addressed for implementation of the theory of discrete-event supervisory control in an operating plant. For example, the events must be generated in real time, based on physical measurements, to provide the supervisor with the current information on the plant; this is beyond what is done off-line for construction of the plant model and control synthesis. Similarly, the event disabling/enabling decisions of the supervisor must be translated in real time as appropriate actions to control the plant.

6.1 *Recommendations for future research*

Synthesis of supervisory control systems may become a significant challenge if some of the events are delayed, intermittent, or not observable at all, possibly due to sensor faults or malfunctions in network communication links. In that case, the control algorithms may turn out to be computationally very complex because of delayed or lost information on the plant dynamics. Future work in this direction should involve research on construction of language measures under partial observation (Chattopadhyay and Ray 2006b) and associated synthesis of optimal control policies to mitigate the detrimental effects of loss of observability. The latter research could be an extension of the work on optimal control under full observation, reported in this paper.

It would be a challenging task to extend the concept of (regular) language measure for languages higher up in the Chomsky hierarchy (Hopcroft *et al.* 2001) such as context-free and context-sensitive languages. This extension would lead to controller synthesis when the plant dynamics is modelled by non-regular languages such as the Petri net (Cassandras and Lafortune 1999, Murata 1989). The research thrust should focus on retaining the polynomial order of computational complexity.

Another critical issue is how to extend the language measure for timed automaton, especially if the events are observed with varying delays at different states. Another research topic that may also be worth investigating is: how to extend the *GF*(2) field, over which the vector space of languages is defined (Ray 2005), to richer fields like the set of real numbers.

Areas of future research also include applications of the language measure in anomaly detection, model identification, model order reduction, and analysis and synthesis of interfaces between the continuously-varying and discrete-event spaces in the language-measure setting. Future research for advancement of the theory of optimal supervisory control for discrete event systems include the following areas:

- Robustness of the control policy relative to unstructured and structured uncertainties in the plant model including variations in the language measure parameters (Lagoa *et al.* 2005)
- Control synthesis under partial observation to accommodate loss of observability at the supervisory level possibly due to sensor faults or communication link failures (Chattopadhyay and Ray 2006b)
- Construction of grammar-based measures, instead of memory-less state-based measures (Chattopadhyay and Ray 2005), for non-regular languages when details of transitions in plant dynamics cannot be captured by finitely many states

Acknowledgement

This work has been supported in part by the U.S. Army Research laboratory and the U.S. Army Research Office under Grant Nos. DAAD19-01-1-0646 and W911NF-06-1-0469.

Appendix A: Derivatives of renormalized measure

This appendix establishes bounds on the derivatives of the renormalized measure $\boldsymbol{\nu}(\theta)$ for all $\theta \in (0,1)$ and computes the limits of the derivatives as $\theta \to 0^+$ as an extension of what was reported in the previous publication (Chattopadhyay and Ray 2006a).

The main result on boundedness of the derivatives of $\nu(\theta)$ are presented as propositions. Specifically, the results reported in Chattopadhyay and Ray (2006a) are combined as the next two propositions.

Proposition A.1: *Let* $\Psi(\theta) \triangleq \theta[\mathbb{I} - (1-\theta)\mathbf{P}]^{-1}$, *where* $\mathbf{P}$ *is a* $(n \times n)$ *stochastic matrix and* $n \in \mathbb{N}$. *Then,*

(i) $\forall\, k \in \mathbb{N}\backslash\{1\}$

$$\lim_{\theta\to 0^+} \frac{\partial^k \Psi(\theta)}{\partial \theta^k} = -k \lim_{\theta\to 0^+} \frac{\partial^k - 1\Psi(\theta)}{\partial \theta^k - 1}[\mathbf{P}+\mathscr{P}][\mathbb{I}-\mathbf{P}+\mathscr{P}]^{-1}$$

(ii) $\lim_{\theta\to 0^+} \frac{\partial^k \Psi(\theta)}{\partial \theta^k}$

$$= \begin{cases} [\mathbb{I}-\mathbf{P}+\mathscr{P}]^{-1} - \mathscr{P}, & \textit{if } k=1 \\ (-1)^k k![\mathbb{I}-\mathbf{P}+\mathscr{P}]^{-\mathbf{1}} & \\ \quad\times\left[\mathbb{I}-[\mathbb{I}-\mathbf{P}+\mathscr{P}]^{-1}\right]^{k-1}, & \textit{if } k \in \mathbb{N}\backslash\{1\}. \end{cases}$$

Proof: Given in Chattopadhyay and Ray (2006a, § 3, pp. 1111–1112 as Corollary 3 and Corollary 6). □

The next proposition establishes bounds on the derivatives of $\boldsymbol{\nu}(\theta)$ in an elementwise sense by computing bounds on the induced sup-norm of the derivatives of $\Psi(\theta)$. Recall that $\boldsymbol{\chi}$ has been defined to have infinity norm equal to 1.

Proposition A.2

$$\left\|\frac{\partial^k \nu(\theta)}{\partial\theta^k}\right\|_\infty \le k!\,2^{k+1}\left(\inf_{\alpha\neq 0}\left\|\left[\mathbb{I}-\mathbf{P}+\alpha\mathscr{P}\right]^{-1}\right\|_\infty\right)^k \quad \forall\theta\in[0,1].$$

Proof: Given in of Chattopadhyay and Ray (2006a, § 3, p. 1113 as Proposition 5). □

Proposition A.3: Denoting the ith element of the kth derivative of the measure vector as $(\partial^k\nu(\theta)/\partial\theta^k)\big|_i$, it follows that

$$\forall k \in \{1,\ldots,n\},\ \frac{\partial^k\nu(\theta)}{\partial\theta^k}\Big|_i = \frac{\partial^k\nu(\theta)}{\partial\theta^k}\Big|_j$$
$$\Longrightarrow \forall\theta\in[0,1], \nu(\theta)\big|_i = \nu(\theta)\big|_j,$$

where n is the number of states in the plant model.

Proof: First it is noted that

$$\nu(\theta) = \theta\sum_{k=0}^{\infty}(1-\theta)^k\mathbf{P}^k\boldsymbol{\chi}\ \forall\theta\in(0,1]$$
$$= \theta\sum_{k=0}^{\infty}\boldsymbol{\Pi}^k(\theta)\boldsymbol{\chi}\ \forall\theta\in(0,1]. \quad (26)$$

Since $\boldsymbol{\Pi}(\theta)$ is a matrix of dimension $n\times n$, it follows from the Cayley–Hamilton Theorem (Bapat and Raghavan 1997) that integral powers of $\boldsymbol{\Pi}(\theta)$ can be expressed as polynomials of degree $n-1$ as follows:

$$\forall r\in\mathbb{N},\ \boldsymbol{\Pi}^r(\theta) = \sum_{k=0}^{n-1} c_k\boldsymbol{\Pi}^k(\theta) \quad \text{with } c_k\in\mathbb{C}. \quad (27)$$

Since each term in the summation on the left hand side of equation (26) is a polynomial in θ of degree $n-1$, it follows that the summation is also a polynomial in degree $n-1$ (since the summation exists due to the sub-stochastic property of $\boldsymbol{\Pi}(\theta)$). Then it follows that each element of $\nu(\theta)$ is a polynomial of degree n. The result then follows from continuity. □

Proposition A.4: For any stochastic matrix $\mathbf{P}$ of dimension $n\times n$, the complexity of computing the limiting matrix $\mathscr{P}$ is of the order of $O(n^3)$.

Proof: Since the limit $\lim_{k\to\infty}(1/k)\sum_{j=0}^{k-1}\mathbf{P}^j = \mathscr{P}$ always exists, it is possible to compute $\mathscr{P}$ within any specified precision simply by computing the sum $\sum_{j=0}^{k-1}\mathbf{P}^j$ followed by division by k, for a large enough value of k. The procedure is summarized in Algorithm 6.

Algorithm 6: Computation of $\mathscr{P}$

```
   input  : $\mathbf{P}$, Desired precision eps
   output: $\mathscr{P}$
 1 begin
 2     Set $Q^{[0]} = \mathbf{0}$ ;                    /* Zero Matrix */
 3     Set $\mathbf{A} = \mathbb{I}$;                  /* Indentity Matrix */
 4     Set $k = 1$;
 5     Set Terminate = false ;
 6     while (Terminate == false) do
 7         $Q^{[k]} = Q^{[k-1]} + \frac{1}{k}\left(\mathbf{A} - Q^{[k-1]}\right)$;
 8         if $\|Q^{[k]} - Q^{[k-1]}\| <$ eps then
 9             Terminate == true ;
10         else
11             k=k+1;
12             A = AP;
13         endif
14     endw
15     $\mathscr{P} = Q^{[k]}$;
16 end
```

Referring to Line 7 of Algorithm 6, it is observed that $Q^{[k]}$ is a stochastic matrix for all k and hence it follows that the algorithm is guaranteed to terminate in (1/eps) iterations, independent of n. Each iteration involves a single matrix multiplication $(\mathbf{P}\times\mathbf{A})$ and hence algorithmic complexity is of the same order as multiplication of two $n\times n$ matrices, i.e., $\le O(n^3)$.

References

R.B. Bapat and T.E.S. Raghavan, *Nonnegative matrices and Applications*, Cambridge, UK: Cambridge University Press, 1997.

C.G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN 0387333320.

I. Chattopadhyay and A. Ray, "A complex measure for linear grammars, *Demonstratio Mathematica*, 38:761–775, 2005.

I. Chattopadhyay and A. Ray, "Renormalized measure of regular languages", *Int. J. Control*, 79, pp. 1107–1117, 2006a.

I. Chattopadhyay and A. Ray, "A language measure for partially observed discrete event systems", *Int. J. Control*, 79, pp. 1074–1086, 2006b.

S.L. Chung, S. Lafortune, and F. Lin, "Supervisory control using variable look-ahead policies", *Discrete Event Dynamic Systems: Theory and Applications*, 4, pp. 237–268, July 1994.

E-E. Doberkat, *Stochastic Automata: Stability, Nondeterminism, and Prediction*, volume 113 of *Lecture Notes in Computer Science*, Springer, Secaucus, NJ, USA, 1981. ISBN 3-540-10835-1.

V.K. Garg, R. Kumar, and S.I. Marcus, "A probabilistic language formalism for stochastic discrete event systems", *IEEE Transactions on Automatic Control*, 44, pp. 280–293, 1999.

V.K. Garg, "An algebraic approach to modeling probabilistic discrete event systems", *Proceedings of 1992 IEEE Conference on Decision and Control*, pp. 2348–2353, Tucson, AZ, December 1992.

V.K. Garg, "Probabilistic languages for modeling of DEDs", in *Proceedings of 1992 IEEE Conference on Information and Sciences*, pp. 198–203, Princeton, NJ, March 1992.

J.E. Hopcroft, R. Motwani and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation, 2nd ed.*, Massachusetts, USA: Addison-Wesley, 2001.

R. Kumar and V.K. Garg, "Control of stochastic discrete event systems modeled by probabilistic languages", *IEEE Transactions on Automatic Control*, 46, pp. 593–606, 2001. URL citeseer.ist.psu.edu/278517.html.

C.M. Lagoa, J. Fu, and A. Ray, "Robust optimal control of regular languages", *Automatica*, 41, pp. 1339–1345, 2005.

M. Lawford and W.M. Wonham, "Supervisory control of probabilistic discrete event systems", in *Proceedings of 36th Midwest Symposium on Circuits and Systems, Detroit, Michigan, USA*, pp. 327–331, 1993.

E.T. Lee and L.A. Zadeh, "Note on fuzzy languages", *Information Sciences*, 1, pp. 421–434, 1969.

R. Milner, *Communication and concurrency*, Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989. ISBN 0-13-115007-3.

M.K. Molloy, "Performance analysis using stochastic petri nets", *IEEE Transactions on Computers*, C-31, pp. 913–917, September 1982.

H. Mortzavian, "Controlled stochastic languages", in *Proceedings of 1993 Allerton Conference*, pp. 938–947, Urbana, IL, 1993.

T. Murata, "Petri nets: Properties, analysis and applications", in *Proceedings IEEE*, 77, pp. 541–540, April 1989.

A. Paz, *Introduction to probabilistic automata (Computer science and applied mathematics)*. Orlando, FL, USA: Academic Press, Inc. 1971. ISBN 0125476507.

S. Phoha, E. Peluso, and R.L. Culver, "A high fidelity ocean sampling mobile network (samon) simulator", *IEEE Journal of Oceanic Engineering, Special Issue on Autonomous Ocean Sampling Networks*, 26, pp. 646–653, 2002.

M.O. Rabin, "Probablistic automata", *Information and Control*, 6, pp. 230–245, 1963.

P.J. Ramadge and W.M. Wonham, "Supervisory control of a class of discrete event processes", *SIAM J. Control and Optimization*, 25, pp. 206–230, 1987.

A. Ray, J. Fu, and C.M. Lagoa, "Optimal supervisory control of finite state automata", *Int. J. Control*, 77, pp. 1083–1100, 2004.

A. Ray, V. Phoha, and S. Phoha, *Quantitative measure for discrete event supervisory control*.Springer, New York, 2005.

A. Ray, "Signed real measure of regular languages for discrete-event supervisory control, *Int. J. Control*, 78, pp. 949–967, 2005.

R. Sengupta and S. Lafortune. An optimal control theory for discrete event systems, *SIAM Journal on Control and Optimization*, 36, pp. 488–541, 1998. URL citeseer.ist.psu.edu/sengupta98optimal.html.

Available online at www.sciencedirect.com

ScienceDirect

Control Engineering Practice 15 (2007) 149–162

CONTROL ENGINEERING PRACTICE

www.elsevier.com/locate/conengprac

Hierarchical control of aircraft propulsion systems: Discrete event supervisor approach

Murat Yasar, Asok Ray*

Mechanical and Nuclear Engineering Department, The Pennsylvania State University, 329 Reber Building, University Park, PA 16802, USA

Received 10 December 2004; accepted 25 May 2006
Available online 13 July 2006

Abstract

This paper presents an application of the recently developed theory of language-measure-based discrete event supervisory (DES) control to aircraft propulsion systems. A two-layer hierarchical architecture is proposed to coordinate the operations of a twin-engine propulsion system. The two engines are individually controlled to achieve enhanced performance and reliability, as necessary for fulfilling the mission objectives. Each engine, together with its continuously varying control system, is operated at the lower level under the supervision of a local discrete-event controller for condition monitoring and life extension; the gain-scheduled feedback controller that is used to maintain the specified performance of the turbofan engine is kept unaltered. A global DES controller at the upper level coordinates the local DES controllers for load balancing and health management of the propulsion system.

Keywords: Discrete event supervisory control; Optimal control; Aircraft propulsion systems

1. Introduction

Discrete-event dynamical behavior of physical plants is often modeled as regular languages that can be realized by finite-state automata (Hopcroft, Motwani, & Ullman, 2001). This paper focuses on development of intelligent decision and control algorithms based on the theory of discrete event supervisory (DES) control (Cassandras & Lafortune, 1999; Ramadge & Wonham, 1987) for a twin-engine aircraft propulsion system.

The DES control system is designed to be hierarchically structured in the following sense. The continuously varying control system of each engine interacts with its own local DES controller for health monitoring and intelligent control; and the operational information is abstracted and reported to the propulsion-level DES controller that coordinates the operation of two engines. Furthermore, the propulsion-level supervisory control system allows interactions with exogenous inputs, such as human operators and inputs from other units (e.g., flight control, structural control, energy management, and avionic systems) of the vehicle management system for flexibility of making on-line modifications in the mission objectives. A good feature of the proposed DES control system is that the supervisory control policy can be adaptively updated on-line at both engine and propulsion levels and that the system is tolerant of small anomalies and component faults.

Although the theory of DES control has been developed for almost two decades (Ramadge & Wonham, 1987), only very few applications have been reported in literature. An apparent reason is that, until recently, no quantitative analytical tool was available for design and evaluation of DES controllers. The work reported in this paper makes use of a quantitative measure of regular languages (Ray, 2005; Ray, Phoha, & Phoha, 2005), and is a novel application of hierarchical DES control synthesis for the nonlinear complex dynamical system of twin-engine aircraft propulsion. The real-time implementation of the DES control scheme is challenging because it requires integration of several disciplines (Kumar & Garg, 1995) such as systems theory, computer hardware and software, and domain knowledge of gas turbine engine propulsion.

*Corresponding author. Tel.: +1 814 865 6377; fax: +1 814 863 4848.
E-mail address: axr2@psu.edu (A. Ray).

0967-0661/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.conengprac.2006.05.011

The DES control of the propulsion system is validated on a simulation test bed. Thus, feasibility of the DES concept is demonstrated for enhanced operation and control of twin-engine aircraft propulsion in the following areas: (i) real-time decision-making for propulsion control (e.g., load balancing between the engines), (ii) damage reduction (with no significant loss of performance) via life extending control, and (iii) enhanced performance and reliability of the mission.

The terms *controller* and *supervisor* are used interchangeably in this paper. The phrases "Upper level" and "Propulsion level" are synonymous and so are the phrases "Lower level" and "Engine level." The paper is organized in six sections including the present one and an appendix. Section 2 describes the real-time simulation test bed of the twin-engine propulsion system. Section 3 presents the syntheses of the engine-level DES control and propulsion-level DES control systems. Section 4 presents the simulation results and discusses implications of the controller design. Section 5 discusses the theoretical and simulation results of performance evaluation for the propulsion system under DES control. The paper is summarized and concluded in Section 6. The appendix provides supporting information and mathematical background for the control policy synthesis in Sections 3 and 5.

2. Description of the test bed for propulsion system simulation

This section presents implementation of DES control on a real-time simulation test bed of a twin-engine aircraft propulsion system, where the engine model is similar, in complexity and details, to that reported by Diao and Passino (2001) and modular aero propulsion system simulation (MAPSS) model (Parker & Guo, 2002). The objective is to validate the theory of optimal DES control for a real-world nonlinear complex dynamical system such as aircraft propulsion.

A DES controlled propulsion system has been designed and implemented on a simulation test bed that consists of three networked computers using the client/server concept. One of the three computers hosts the propulsion system coordinator for health monitoring of the engines and accordingly making intelligent decisions (e.g., load balancing). The other two computers run separate copies (which may or may not be different depending on the health of the individual engines) of the gas turbine engine simulation model including its continuously varying gain-scheduled feedback control system and a local discrete-event supervisor. The test bed is capable of simulating different dynamics for individual engines due to non-uniform operating conditions. Each of the engine simulators integrates the event-driven discrete dynamics modeled by finite-state automaton as well as time-driven continuous dynamics modeled by ordinary differential equations through continuous-to-discrete and discrete-to-continuous interfaces (Fu, Yasar, & Ray, 2004). Fig. 1 shows the supervisory control architecture of the engine propulsion control system. This software architecture is flexible to adapt deterministic finite state automaton (DFSA) models and controller designs for other complex dynamical systems. Each major function in the simulation program has a modular structure as implemented on the three networked computers of the simulation test bed.

The C+ + code of the multi-layer DES control system is superimposed on the existing FORTRAN simulation code of the turbofan engine model and the associated continuously varying engine control system through a C+ + program. Specifically, the wrapper program interfaces the major inputs and outputs of the FORTRAN simulation code with the rest of the program in the C+ + environment. This approach takes advantage of the available FORTRAN models as individual modules of the integrated C+ + program without making any significant changes.

The FORTRAN code of the turbofan engine simulation program, which consists of high-order nonlinear differential and difference equations and supporting algebraic

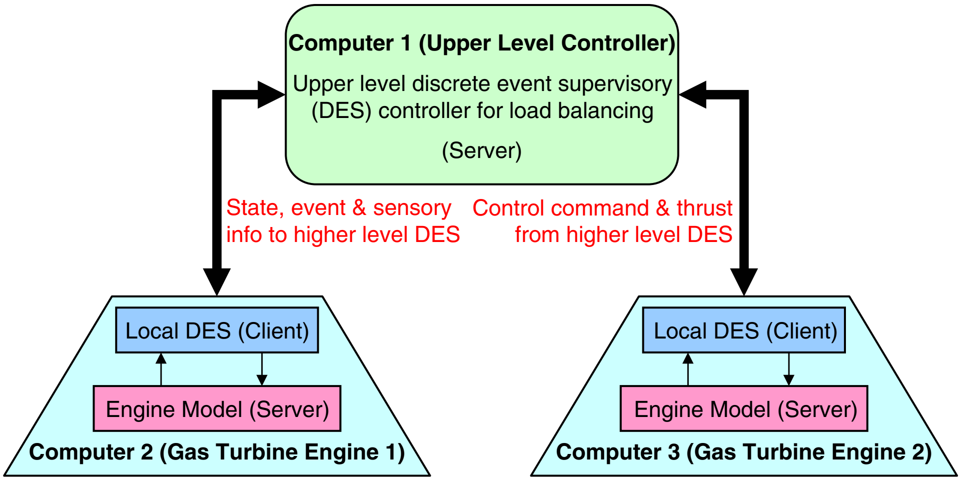

Computer 1 (Upper Level Controller)
Upper level discrete event supervisory (DES) controller for load balancing
(Server)
State, event & sensory info to higher level DES
Control command & thrust from higher level DES
Local DES (Client)
Engine Model (Server)
Computer 2 (Gas Turbine Engine 1)
Local DES (Client)
Engine Model (Server)
Computer 3 (Gas Turbine Engine 2)

Fig. 1. Supervisory control architecture of the propulsion system.

equations, has been designed for both steady-state and transient operations of a generic jet engine (Diao & Passino, 2001; Parker & Guo, 2002); and different control strategies have been reported in the literature (Diao & Passino, 2001; Fu et al., 2004; Litt, Parker, & Chatterjee, 2003). With the proper inputs such as power lever angle (PLA), and operating condition parameters (e.g., altitude, forward speed, and ambient temperature), the FORTRAN engine model simulates the engine operations from a steady state to transients and to (possibly) other steady states. This simulation code is a stand-alone program with a gain-scheduled feedback controller. The engine simulation model provides various sensor data (e.g., combustion-chamber temperature and high-pressure and low-pressure turbine speeds) together with other critical information (e.g., simulation step size and simulation cycle number), which are collected by the C++ wrapper program and exchanged with the DES controllers through a typical message application protocol interface (API) communication routine. This communication protocol sends and receives message packages through TCP and/or UDP networks. The typical delay in this protocol interface is mainly due to the network communications and is found to be less than 1 ms. Since the engine and flight simulations use integration step sizes in the order of 10–20 ms, the communication delays do not have a major bearing on performance of the proposed control architecture.

Fig. 2 shows the architecture of the engine-level plant and DES controller implementation, which has two replicas, with possible different parameters and initial conditions, in two different computers. Fig. 3 shows the organization of the propulsion-level DES controller together with its own (discrete) event generator, which is implemented on a third computer and makes use of the Message API to communicate with the other two computers.

The DES controller design has two important components that serve as interfaces between the continuously varying control system and the discrete-event supervisory controller—one is event generator and the other is action generator. Event generator receives continuously varying sensor data from the engines. The data along with other information like estimated state and external inputs are used to generate events that, in turn, are inputs to the unsupervised DFSA model of engine operation. The DFSA model is constructed based on the operation scenario; the details are discussed later in Section 3. The DFSA model also serves as a state estimator and provides information on engine states and (both controllable and uncontrollable) events for the discrete-event supervisor to take appropriate actions. Event behavior in the state-based DFSA model is dependent on the state where the event is generated and not on the history or the path of how the state is reached.

The DES controller represents the control policy applied to the DFSA model of engine operation, and it could be a conventional DES controller based on the control specifications (Ramadge & Wonham, 1987) provided by an experienced designer; alternatively, an optimal discrete-event supervisor can be designed by the quantitative method (Fu et al., 2004; Ray, Fu, & Lagoa, 2004). In both cases, the DES controller takes the estimated states as inputs and generates control commands (of controllable event disabling or enabling) as outputs, which are transmitted through a Message API communication routine to the action generator. The primary task of the action generator is to convert the control commands from the supervisor into necessary input functions for the continuously varying plant.

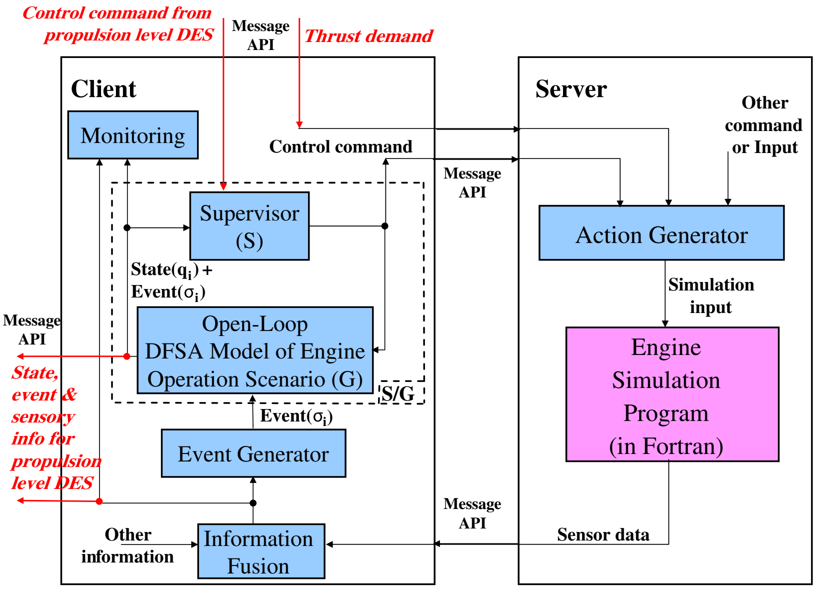


Fig. 2. Engine level plant/DES controller implementation.

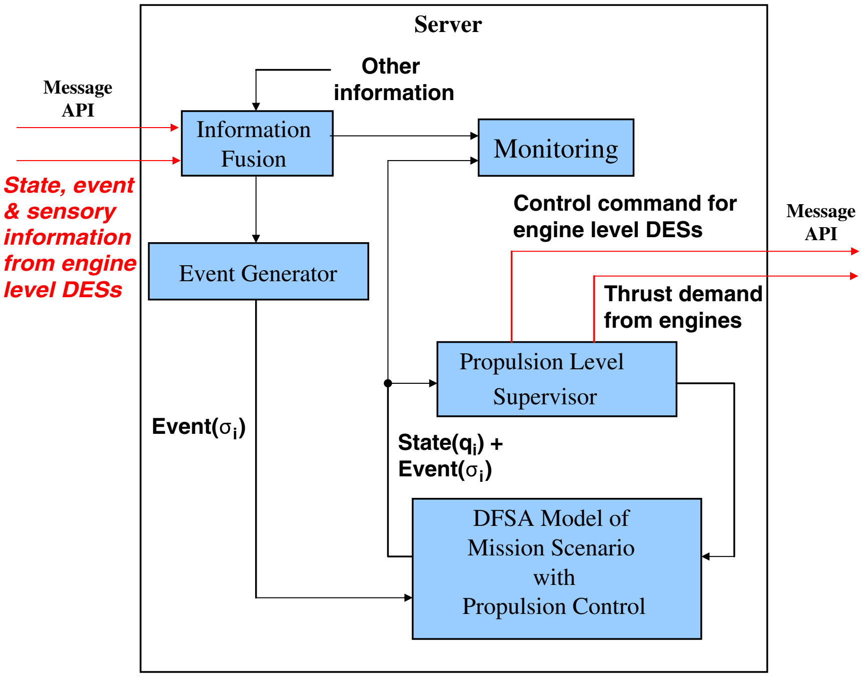

Server
Message API
Other information
Information Fusion
Monitoring
State, event & sensory information from engine level DESs
Event Generator
Control command for engine level DESs
Message API
Thrust demand from engines
Propulsion Level Supervisor
Event(σ_i)
State(q_i) + Event(σ_i)
DFSA Model of Mission Scenario with Propulsion Control

Fig. 3. Propulsion level DES controller implementation.

3. Synthesis of DES controllers

One of the major tasks of the supervisory decision making is fusion of the (possibly) redundant, conflicting, and incomplete information to make timely decisions. Such information can be derived from different types of sensor data (Volponi, Brotherton, Luppold, & Simon, 2003) as well as operational history and the knowledge-base generated from pilot's personal experience. Computer-based advanced analytical techniques are necessary for fusion of the time series data available from multiple sensors and other relevant non-sensor-based information to make specific inferences that could not be achieved through the sole usage of the available sensory information. Improved performance may not result simply from an increased volume of sensor data and engine information unless the ensemble of information is systematically processed in the context of the engine operational conditions and mission objectives (Tolani, Yasar, Ray, & Yang, 2006). In order to achieve the desired performance of a DES controller, it is essential to have an effective event generation algorithm to ensure fusion of the heterogeneous information for: (i) enhanced resolution and reduced ambiguity in decision and control; and (ii) advantageous trade-offs between probability of false alarms and missed detection (Basseville & Nikiforov, 1993).

The unsupervised dynamics of engine operations are modeled as a DFSA, based on postulated scenarios. (Note that the model may change for different mission objectives.) In the present context, the DFSA model assumes that a twin-engine aircraft is carrying out a routine surveillance mission. The mission abortion is allowed at certain states according to the operation scenario. Each engine of the aircraft is equipped with a continuously varying controller which is supervised by a local engine-level DES controller. The primary objective of the local DES controller is to strike the right balance between the conflicting demands of higher performance from upper level supervisor and limiting structural damage to the engine components. The global propulsion-level DES controller redistributes the load depending on the health of the engine and thrust demand placed by the pilot.

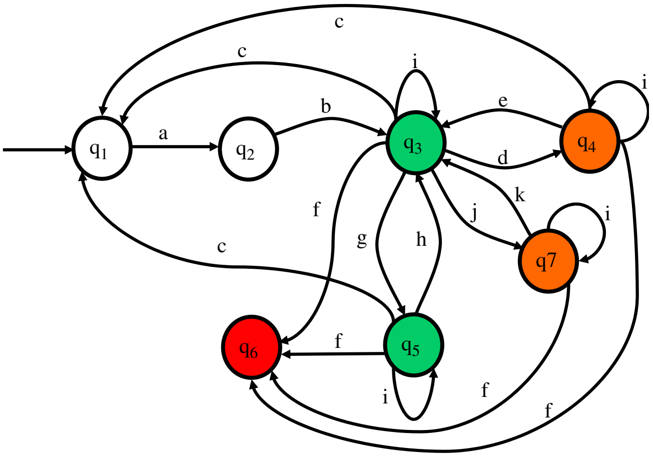


Fig. 4. Unsupervised plant DFSA model at the engine level.

3.1. Engine-level DES control

Fig. 4 presents the DFSA model of the unsupervised engine operations for DES control. Table 1 lists the events, where "C" denotes controllable events and "UC" denotes uncontrollable events; and Table 2 lists the plant states.

Table 1
Event list for the unsupervised engine model

Event	Description	Status
a	Start	C
b	Warm up complete	UC
c	Shut the engine	UC
d	Detection of oscillations	UC
e	Nozzle area reduction	C
f	Engine fails	UC
g	Reduce performance/reduce damage	C
h	Increase performance/increase damage	C
i	Remain in the state	C
j	Reduce performance	C
k	Increase performance	C

Table 2
State list for the unsupervised engine model

State	Description	Status
$\mathbf{q}_1$	Engine start	
$\mathbf{q}_2$	Engine warm up	
$\mathbf{q}_3$	High performance/High damage rate	Marked (good)
$\mathbf{q}_4$	Oscillations	Marked (bad)
$\mathbf{q}_5$	Low performance/Low damage rate	Marked (good)
$\mathbf{q}_6$	Engine inoperable	Marked (bad)
$\mathbf{q}_7$	Low performance/High damage rate	Marked (bad)

The events and states for the DFSA models at the engine level are denoted by lower case letters (e.g., **a** is the start event and $\mathbf{q}_1$ is the engine start state). The engine can operate in two regimes, one is high-performance regime (state $\mathbf{q}_3$), where the damage rate is also high. The other is low-performance regime (state $\mathbf{q}_5$), where the damage rate is low. In the high-performance regime, the engine has a tendency to go to state $\mathbf{q}_4$, where engine variables like combustor temperature have been observed to have oscillatory behavior. Temperature oscillations could be extremely harmful for engine health (DeLaat & Chang, 2003) and must be avoided at all costs. Engine-level controller chooses the regime of operation (state $\mathbf{q}_3$ or $\mathbf{q}_5$) depending on two factors: thrust requirement at the propulsion level and health of the engine, as explained below.

Health of the engine is determined from the damage accumulation that is a function of high-pressure turbine gas inlet temperature and shaft speed; and in addition, damage spikes (i.e., sudden jumps) at random time intervals are introduced to simulate the damage in a real-world engine. The DFSA model of the supervised engine operations is shown in Fig. 5, where the dashed lines indicate those controllable events that are disabled by the optimal control algorithm, described in the Appendix and citations therein. The state $\mathbf{q}_7$ in Fig. 5 becomes unreachable following the disabling action of the supervisor. Therefore, all transitions, originating from the state $\mathbf{q}_7$, are also shown with dashed lines as well as the state itself.

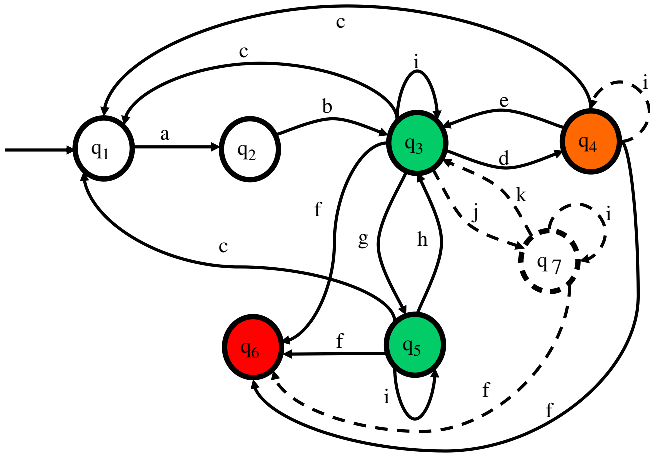


Fig. 5. Supervised plant DFSA model at the engine level.

Table 3
Event list for the propulsion level DFSA model

Event	Description	Status
A	Start engines	C
B	Warm up complete	UC
C	One engine deteriorates	UC
D	Redistribute the load (one engine is unhealthy)	C
E	Both engines deteriorate	UC
F	One good engine fails	UC
G	Both engines fail	UC
H	Increase performance	C
I	Reduce performance	C
J	Request to abort mission	C
K	Request accepted	UC
L	Request rejected (both engines are running)	UC
M	Mission accomplished	UC
N	Turn off engines	UC
O	Redistribute load (both engines are unhealthy)	C
P	Redistribute load (one engine has failed)	C
Q	Request rejected (one engine has failed)	UC
R	One bad engine fails	UC

3.2. Propulsion-level DES control

One of the main tasks of the propulsion-level DES controller is to redistribute the load between two engines depending on the current health condition of each engine and the thrust demand. A DES controller is expected to ensure that this requirement is satisfied, regardless of being optimal or not. The propulsion-level DES controller acts in an advisory role for the mission-related decisions to enhance the mission performance. However, the ultimate decision for mission-related operations is left for the pilot.

Since the propulsion-level supervisor is designed to execute the key decisions at the engine level, the model for operating regimes of the propulsion system include the Cartesian product of the state sets of two (locally supervised) engine models. However, model order reduction via aggregation and deletion of unrealizable states is

Table 4
State list for the propulsion level DFSA model

State	Description	Status
$\mathbf{Q}_1$	Aircraft on ground	
$\mathbf{Q}_2$	Engines warming up	
$\mathbf{Q}_3$	Both engines in high performance operation	
$\mathbf{Q}_4$	One engine in high one engine in low performance	
$\mathbf{Q}_5$	Both engines in low performance operation	
$\mathbf{Q}_6$	One engine stopped one engine in high performance	
$\mathbf{Q}_7$	One engine stopped one engine in low performance	
$\mathbf{Q}_8$	Both engines failed	Marked (bad)
$\mathbf{Q}_9$	Decision for abort mission	Marked (bad)
$\mathbf{Q}_{10}$	Mission successful	Marked (good)
$\mathbf{Q}_{11}$	High damage detected for one engine	
$\mathbf{Q}_{12}$	High damage detected for both engines	

needed because the above Cartesian product may result in a very large number of states. The events and states of the unsupervised DFSA model at the propulsion level are listed in Tables 3 and 4, respectively. Events and states for the DFSA model at the propulsion level are denoted by upper case letters (e.g., **A** is the start event and $\mathbf{Q}_1$ is the aircraft on the ground state). Multiple occurrences of an event have been indicated as single events as seen in three instances (i.e., events **D**, **O** and **P**) in Table 3. The event "request to abort mission" is controllable while the events such as "request accepted", "abort the mission" and "shut down the engine" are uncontrollable events from the supervisor's perspectives.

4. Simulation experiments: results and discussion

Experiments were conducted on the simulation test bed, described in Section 2, to validate the DES control concept. Upon successful implementation of the software modules on the client and server computers, several simulation experiments were performed. The first set of experiments was performed at the engine level to demonstrate the interactions between the DES controller and continuous-time dynamics of the engine. The design specifications of the engine-level supervisor include reduction of the engine component damage and consequently engine life enhancement. Then, the effects of the propulsion-level DES controller that is built upon the engine-level DES controllers are investigated.

Fig. 6 exhibits the predetermined input profile that excites both unsupervised and optimally supervised engine models. Time responses of several outputs (combustor outlet temperature, high-pressure turbine speed, net thrust of the engine, and fuel flow into the main burner) over a period of 12 min were observed. The four plates in each of Figs. 7 and 8 exhibit the response profiles of the above set of plant variables for unsupervised and supervised engine models, respectively. A comparison of plots in Figs. 7 and 8 indicates that the optimal DES control at the engine level eliminates the high frequency oscillations that are present in the unsupervised plant responses in Fig. 7. The supervisor takes actions immediately upon detection of oscillations by an FFT algorithm. Without making any alterations in the gain-scheduled controller of the engine, the supervisory actions are implemented as a piecewise constant term in the reference input to the (continuous gain-scheduled) controller. In the continuous control system, there are seven summation points for the feedback loop, each providing a reference signal for a specific actuator. It is found that the booster vane angle and nozzle area manipulations have the most significant effects on the response of engine variables (Tolani et al., 2006). In the work reported in this paper, nozzle area is decreased by 20% of the nominal value. Due to supervisor's actions, the potentially sustained oscillations are quenched in less than 30 s after oscillations are observed, and the engine operation is brought to steady state. Thus, sustained oscillations are practically non-existent in the supervised plant responses in Fig. 8. Since high-frequency oscillations of temperature and pressure are the primary sources of fatigue crack damage in the turbine blades, disks, and stationary vanes, the supervisory control becomes very effective for mitigation of structural damage in the engine components. In contrast, in the unsupervised case, the engine health would be adversely affected if the propulsion system is operated in this way to achieve the mission objectives.

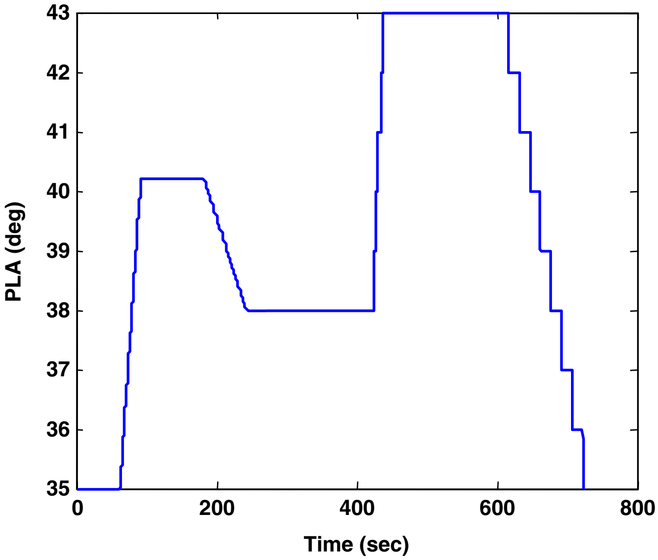


Fig. 6. Power lever angle input.

The propulsion-level DES controller has three main tasks. The first task is the intelligent decision making and control of the twin-engine aircraft propulsion systems for mission execution; the second task is to improve the overall mission and operational behavior so that engine health can

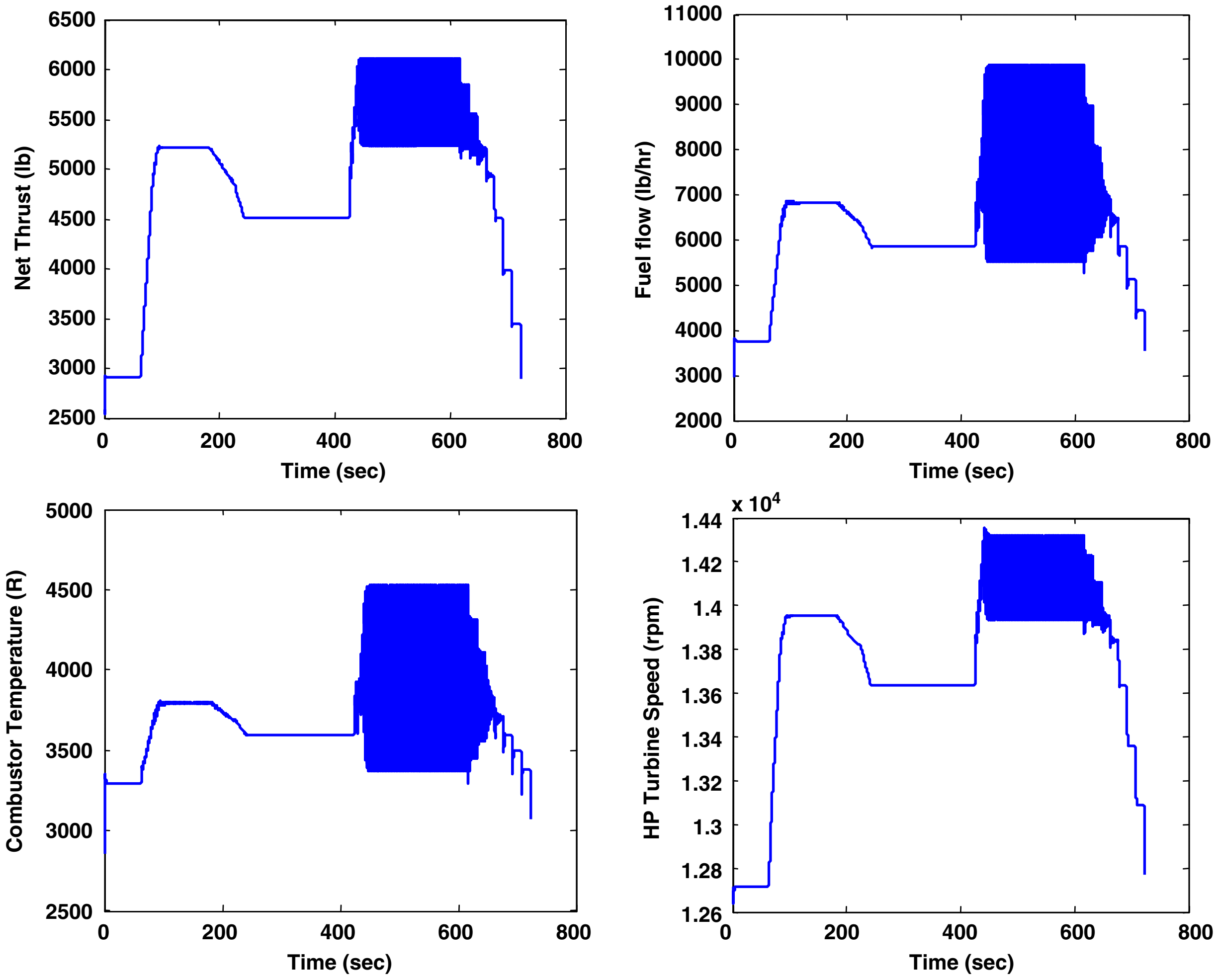


Fig. 7. Simulation output for the unsupervised case.

be enhanced via damage reduction; and the third task is load balancing between two engines so that the propulsion system produces the thrust demanded by the flight control system while attempting to enhance engine life. The issue of load balancing becomes even more important when the health conditions of two engines are significantly different (one can be in "bad condition" and the other in "good condition"). If the situation comes to this point, then the aim of the DES controller is judicious redistribution of the load between two engines such that the "bad engine" carries lower load than the "good engine", subject to the condition that the total thrust output of the engines satisfies the mission and safety requirements. To satisfy these design requirements, the propulsion-level DES controller (Kumar & Garg, 1995; Ramadge & Wonham, 1987), which may or may not be optimal, is designed and implemented over the unsupervised plant model.

Figs. 9 and 10 show the simulated outputs of two engines, where both engines are in "good" health at the start of simulation. The damage increment is a dimensionless quantity denoting the damage accumulation as percentage of the nominal value at each simulation cycle and is attributed to the turbine blades. As the mission progresses, due to an injected fault in Engine 2, the engine deteriorates and starts to run at "bad" health condition after 330 s. After 630 s, another fault is injected in Engine 1 and this engine also deteriorates. Thus, the load distribution of the engines varies in three regions (see Figs. 9 and 10). In the beginning, the load is equally distributed in Region 1. Then the "good engine" (Engine 1) takes the responsibility of producing higher thrust as seen in Region 2 in the plates of Figs. 9 and 10. Later on, both engines are again loaded equally as seen in Region 3 when it is not advisable to impose uneven thrust requirements. The first damage event affects the performance of Engine 2 after approximately 50 s in order to realize the thrust demand of the pilot. At this point, the full thrust demand cannot be satisfied if the load in Engine 2 is reduced. While the thrust produced in Region 2 by each engine is not the same, the most critical impact of the uneven load distribution is formation of excess moment along the yaw axis of the aircraft. However, it might be possible to counteract this situation by adjusting the control surfaces of aircraft (Yasar, Horn, & Ray, 2006).

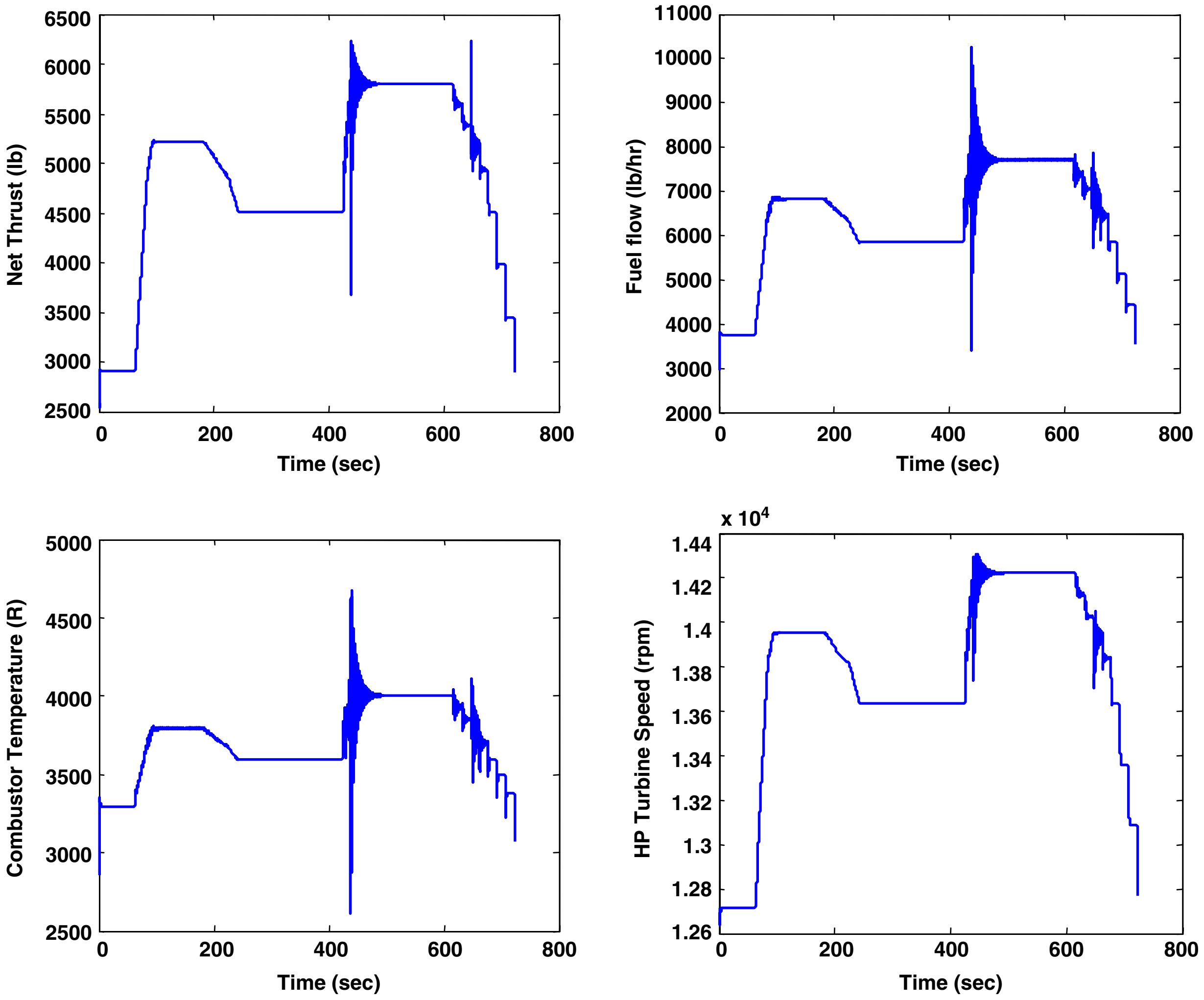


Fig. 8. Simulation output for the optimally supervised case.

5. Evaluations of DES controllers

The language measure, described in Ray (2005) and Ray et al. (2005), has been used to quantitatively evaluate the impact of the propulsion-level DES controller on the overall mission behavior. Given the state weights and the state transition probabilities, the language measure provides a quantitative performance measure of the controllers. Supporting information and pertinent mathematical background are given in the Appendix.

5.1. Identification of state transition probabilities

Quantitative analysis of DES controllers and synthesis of an optimal DES controller require identification of the state transition cost matrix. Similar to continuously varying dynamical systems (CVDS), one must use the techniques of system identification (Ljung, 1999) to evaluate the parameters of the unsupervised DFSA plant model, i.e., the elements $\tilde{\pi}^0_{ij}$ of the event cost matrix $\tilde{\Pi}^0$ (see Definition 2 in the Appendix). As the number of experiments increases, the identified event costs tend to converge within an appropriately specified tolerance. For stationary operation of the engine, since conditional probabilities of the events are assumed to be time-invariant, the identified event costs and their uncertainty bounds can be determined. The probabilities of the events such as deterioration and failure of an engine are triggered by the event generation algorithm, based on the sensory information. Nevertheless, it is not possible to simulate some events, such as acceptance or rejection of the request by the pilot, without a random event generator. The randomization used in triggering this type of events has certainly an effect on the identified event costs, but not directly, since sensor-based events are not induced by this randomization (Ray et al., 2005). As a typical case, Fig. 11 presents identification of event probabilities at state $\mathbf{Q}_5$, where both engines are in low-performance operation.

Fifty simulated missions were performed for both unsupervised plant G and supervised plant S/G to construct the respective state transition probability matrices, Π^0 and Π^S. The different visited states and the triggered events were monitored and plotted to obtain the particular $\tilde{\Pi}$-matrices. After identifying the respective

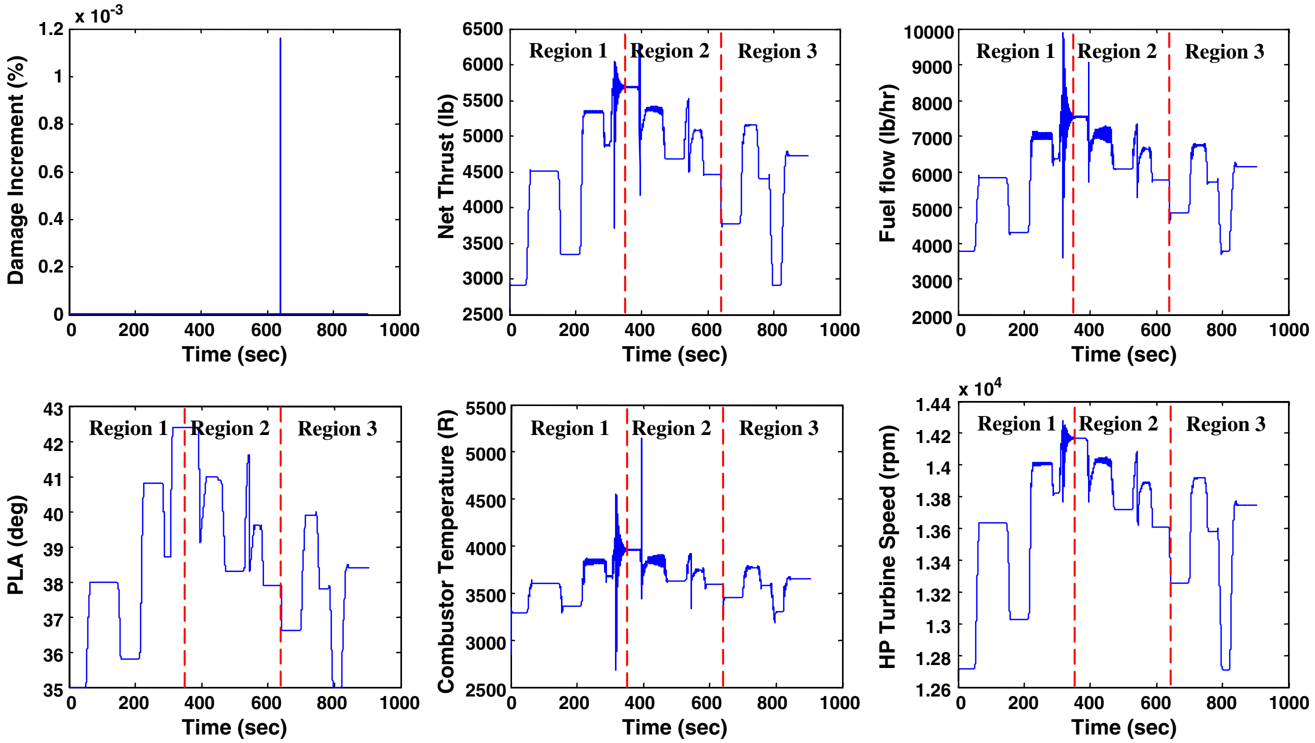


Fig. 9. Effect of conventional propulsion level DES controller on engine 1.

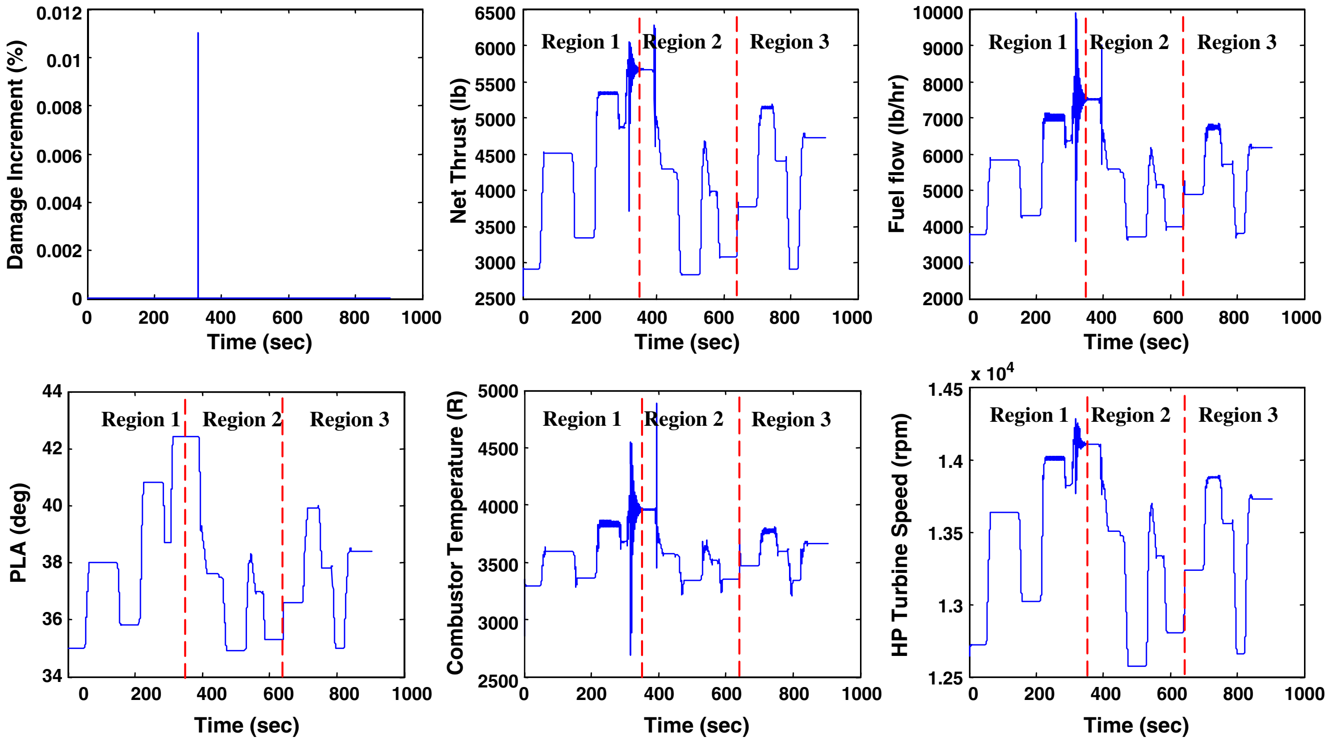


Fig. 10. Effect of conventional propulsion level DES controller on engine 2.

event probability matrices, $\tilde{\Pi}^0$ and $\tilde{\Pi}^S$, of G and S/G, the state transition probability matrices Π^0 and Π^S are computed using Definition 3 in the Appendix. Table 5 lists the Π^0-matrix of the unsupervised DFSA model at the propulsion level.

5.2. *Selection of characteristic values for evaluation of DES controllers*

Given the state characteristic values and the state transition probabilities, the language measure serves

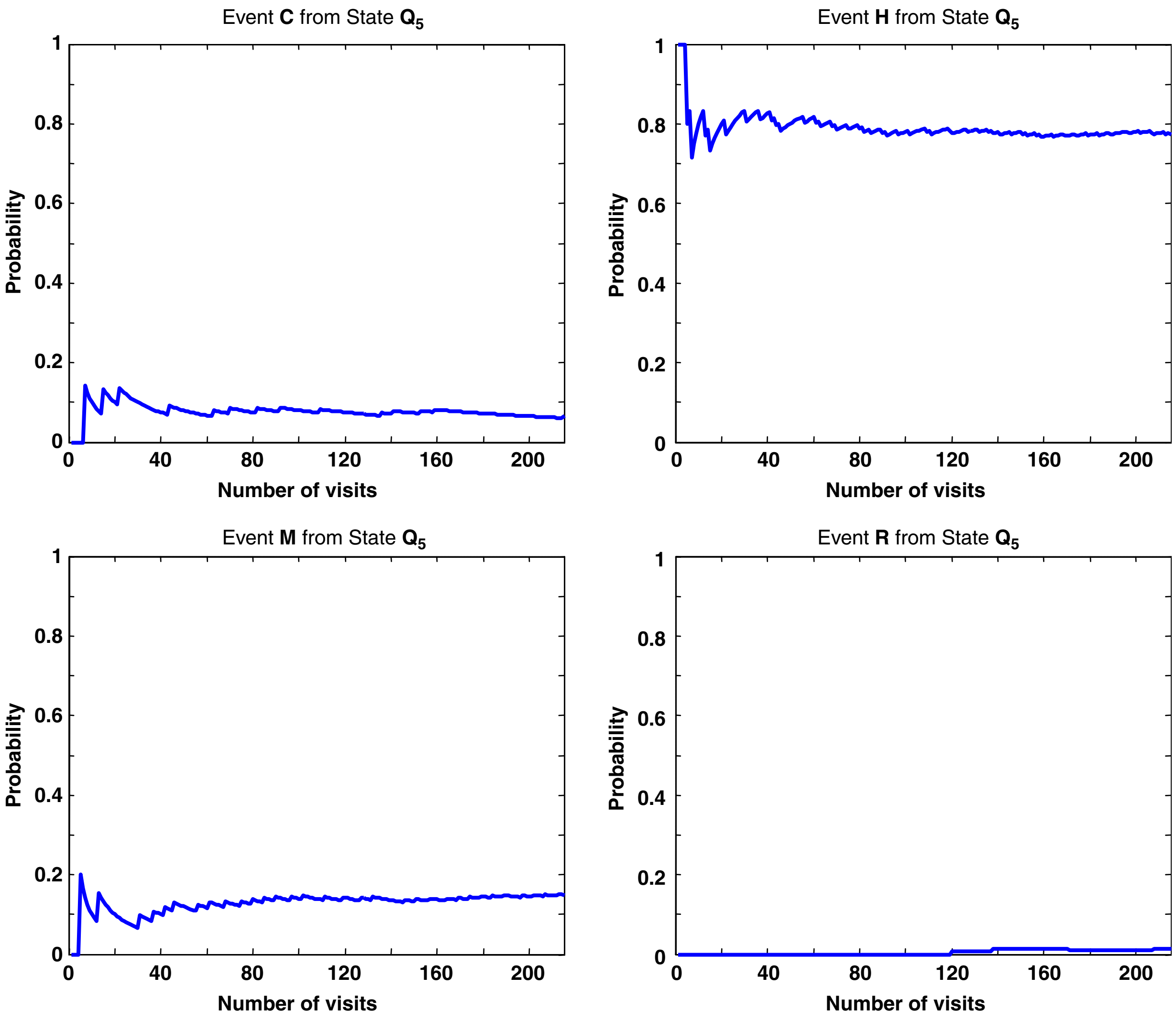


Fig. 11. Convergence of event cost identification.

Table 5
Π matrix of the propulsion level DFSA model

State	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}	Q_{11}	Q_{12}
Q_1	0	1	0	0	0	0	0	0	0	0	0	0
Q_2	0	0	1	0	0	0	0	0	0	0	0	0
Q_3	0	0	0	0	0.899	0.037	0	0	0	0	0.051	0.014
Q_4	0	0	0.500	0	0.500	0	0	0	0	0	0	0
Q_5	0	0	0.772	0	0	0	0.014	0	0	0.149	0.065	0
Q_6	0	0	0	0	0	0	0.250	0.125	0.625	0	0	0
Q_7	0	0	0	0	0	0	0	0	1	0	0	0
Q_8	0	0	0	0	0	0	0	0	0	0	0	0
Q_9	0.415	0	0	0	0	0	0	0	0	0	0.512	0.073
Q_{10}	1	0	0	0	0	0	0	0	0	0	0	0
Q_{11}	0	0	0	0.044	0.370	0	0	0	0.587	0	0	0
Q_{12}	0	0	0	0	0.333	0	0	0	0.667	0	0	0

as a theoretical performance measure for quantitative evaluation of DES controllers. The characteristic values are assigned based on the designer's perception for the importance of terminating on specific marked states. For the propulsion-level DES controller, the weights of the states are selected according to each state's importance (contribution) to the mission management as

$$\bar{\chi} = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ -1.00\ \ -0.188\ \ 0.300\ \ 0\ \ 0].$$

The non-marked states have no direct bearing on the mission performance and hence each state (i.e., states $\mathbf{Q}_1$ to $\mathbf{Q}_7$, $\mathbf{Q}_{11}$ and $\mathbf{Q}_{12}$) has been assigned zero weight. The marked states, $\mathbf{Q}_8$, $\mathbf{Q}_9$ and $\mathbf{Q}_{10}$, which do have direct bearing on the mission performance, are assigned non-zero weights as follows:

$\mathbf{Q}_8$: *Both engines failed*, State $\mathbf{Q}_9$: *Mission abort*, State $\mathbf{Q}_{10}$: *Mission successful*. They have relative weights of -1, -0.188 and 0.3, respectively. The bad marked state, $\mathbf{Q}_8$, is assigned the characteristic value of -1.0 because the aircraft will most likely be destroyed (because of both engines being non-functional) if the DFSA terminates on this state. On the other hand, the good marked state, $\mathbf{Q}_{10}$: *Mission successful*, is assigned the characteristic value of $+0.3$ based on its relative importance to the loss of the aircraft. Therefore, by choosing the characteristic values in this way, loosing one aircraft is made equivalent to approximately three successful missions. The other bad marked state, $\mathbf{Q}_9$: *Mission abort*, has also a negative characteristic value which signifies the importance of this state relative to a successful mission and a possible loss of the aircraft. The selection of characteristic value for mission abort is to quantitatively match the theoretical (i.e., language measure-based) performance of the unsupervised plant with its experiment-based performance (which will be introduced in the next section). The corresponding performance measures for the supervised cases must also match with this selection provided that the plant is modeled appropriately and the language measure parameters, Π-matrix and $\bar{\boldsymbol{\chi}}$-vector, are correctly assigned.

Using the relation $\bar{\boldsymbol{\mu}} = [I - \Pi]^{-1}\bar{\boldsymbol{\chi}}$ derived in Ray (2005), the language measures (i.e., the theoretical performance of the propulsion system) are calculated for unsupervised plant model G and DES controlled plant S/G as $\mu_{unsupervised} = 0.1260$ and $\mu_{supervised} = 0.2055$, respectively. It is seen that the DES controller used in the propulsion level has a positive effect on the mission behavior of the propulsion system.

5.3. Optimal DES controller synthesis and evaluation

After identifying the event cost matrix $\tilde{\Pi}^0$ of the unsupervised plant G, the state transition cost matrix Π^0 is constructed (using Definition 3 in the Appendix). State transition cost matrix is the only unknown input for the optimal control algorithm to design the optimal DES controller, since the other necessary parameters, such as the characteristic vector $\bar{\boldsymbol{\chi}}$, are selected by the designer based on the design requirements. Therefore, given the state transition cost matrix Π^0 of the unsupervised plant and the state characteristic vector $\bar{\boldsymbol{\chi}}$, the optimal DES controller can be synthesized as described in the Appendix.

Table 6
Iterations for optimal DES controller synthesis

State	Unsupervised plant	Iteration 1	Iteration 2
$\mathbf{Q}_1$	0.1260	0.2452	0.2452
$\mathbf{Q}_2$	0.1273	0.2477	0.2477
$\mathbf{Q}_3$	0.1286	0.2502	0.2502
$\mathbf{Q}_4$	0.1415	0.2617	0.2617
$\mathbf{Q}_5$	0.1572	0.2785	0.2785
$\mathbf{Q}_6$	**−0.2555**	−0.1238	−0.1238
$\mathbf{Q}_7$	**−0.1510**	*0.0000*	0.0000
$\mathbf{Q}_8$	**−1.0000**	−1.0000	−1.0000
$\mathbf{Q}_9$	**−0.1525**	−0.0353	−0.0353
$\mathbf{Q}_{10}$	0.4248	0.5428	0.5428
$\mathbf{Q}_{11}$	**−0.0250**	*0.1132*	0.1132
$\mathbf{Q}_{12}$	**−0.0488**	*0.0919*	0.0919

Table 7
Language measure and performance for propulsion level supervisors

	Unsupervised	Supervised	Optimal
Language measure ($\boldsymbol{\mu}$)	0.1260	0.2055	0.2452
Performance Index ($\boldsymbol{\nu}$)	0.1276	0.1959	0.2480

Table 6 lists the iterations of optimal control synthesis for the propulsion-level supervisory control where the first column belongs to the unsupervised plant G. The performance measure of the unsupervised plant is negative at the states $\mathbf{Q}_6$, $\mathbf{Q}_7$, $\mathbf{Q}_8$, $\mathbf{Q}_9$, $\mathbf{Q}_{11}$, and $\mathbf{Q}_{12}$ as indicated by the bold scripts in Table 6. All controllable events leading to these states are disabled and the resulting performance measure at Iteration 1 shows sign change at states $\mathbf{Q}_7$, $\mathbf{Q}_{11}$, and $\mathbf{Q}_{12}$ as indicated by italics in Table 6. All controllable events leading to these states are now re-enabled for further increase in performance at Iteration 2. However, there is no sign change in the performance vector between Iteration 1 and Iteration 2, which immediately shows that the algorithm converged to the optimal solution after this iteration. The synthesis is complete in Iteration 2 (i.e., there is no need to go for the Iteration 3) because there is no sign change; moreover, the performance vector at Iteration 2 shows also no improvement after the previous iteration.

The performance of the optimal controller was compared with that of unsupervised plant G and the supervisor S that was designed using the conventional procedure (Cassandras & Lafortune, 1999; Ramadge & Wonham, 1987). Theoretical performance of the supervisors can be associated with the language measure of each supervisor, as described in the Appendix. The language measures of the unsupervised plant and conventional and optimally supervised plants at the propulsion level are listed in Table 7 that also shows a close agreement between the analytically generated language measures and experimentally determined values.

Results of simulation experiments have been used to validate the DES controller performance based on the language-theoretic analysis. The experimental performance index is determined as a function of the relative weights of the visited states. The mission outcomes of the unsupervised and supervised propulsion systems were recorded during each simulated mission. The numbers of missions ending at both good and bad marked states were multiplied by the respective relative weights of the states. That is, the experimental performance measure is calculated as

$$v = \frac{\sum_{i=1}^{N} n_i \cdot \chi_i}{N},$$

where n_i is the number of missions ending at the ith state, and N is the total number of experiments; and χ_i is the characteristic weight of the ith state. The performance of the unsupervised plant and the other two supervisors, namely, conventionally designed supervisor (Ramadge & Wonham, 1987) and language-based optimal supervisor (Ray et al., 2004), are compared based on the observations of mission execution on the simulation test bed. The experimental evaluations of the performance for different supervisors are presented in Table 7. Both theoretical and experimental (simulation) evaluations of DES controllers provide better mission management under optimal supervision. It is seen that the theoretical performance of the supervisors is in quantitative agreement with the experimental results, presented in Table 7. Optimal DES controller, synthesized using the algorithm described in Appendix and in Ray et al. (2004), has the highest theoretical performance (i.e., highest language measure) among all controllers. The results of the simulation experiments concur with the theoretical measure of the controllers in the sense that optimal supervisor yields the best mission performance.

Remarks on the optimal supervisor design: At the first step of the iterations, the performance measure of the unsupervised plant is negative at states $\mathbf{Q}_6$, $\mathbf{Q}_7$, $\mathbf{Q}_8$, $\mathbf{Q}_9$, $\mathbf{Q}_{11}$, and $\mathbf{Q}_{12}$. Therefore, controllable transitions to these states should be disabled. It is observed that one engine is lost at the states $\mathbf{Q}_6$ and $\mathbf{Q}_7$, implying that the events leading to these states would cause the "one engine failed" condition which is uncontrollable and hence cannot be disabled. Similarly, the only transition to the state $\mathbf{Q}_8$ causes the "both engines failed" condition, which is also uncontrollable; hence, this transition cannot be disabled either. The transitions leading to states $\mathbf{Q}_{11}$ and $\mathbf{Q}_{12}$ are the "deterioration of engines" condition, which is directly related to the damage information received from the engine-level DES control system. Based on this information, the propulsion-level supervisor makes a decision on the health condition of the individual engines. Evidently, this kind of sensory events are uncontrollable, so is the engine deterioration event. The remaining state that should be investigated is the state $\mathbf{Q}_9$, *Mission abort*. The optimal control algorithm disables all mission abort requests, which significantly increases the performance of the mission behavior with increased risk of losing the aircraft. The simulation experiments show that the mission performance at the propulsion level increases under the optimal DES controller, albeit with an increased probability of aircraft loss. Table 7 shows a close agreement between the analytically generated language measures and experimentally determined performance data. It should be noted that, the optimal control policy is likely to change if the elements (i.e., individual state weights) of the $\bar{\chi}$-vector are altered.

6. Summary and conclusions

This paper presents a quantitative approach to analysis and synthesis of hierarchical DES control laws for aircraft propulsion systems. The objectives are:

- Intelligent decision and control of distributed propulsion management systems, where each of the engines has its own local DES control.
- Structural damage reduction and life extension of aircraft engines without any significant loss of the system performance.
- Decision making and mission planning modifications through a high-level DES coordinator.
- Incorporation of optimal control laws for enhanced mission management.
- Extension of this work to other complex dynamic systems such as rotorcrafts and power plants simulation test beds.

A decision and control architecture has been proposed to coordinate the operations of a twin-engine propulsion system. The DES control law has been validated for a twin-engine aircraft propulsion system on a networked simulation test bed. The plant dynamics in the simulation test bed is built upon the model of a generic turbofan gas turbine engine. The software architecture of the simulation test bed is flexible for adaptation to arbitrary DFSA models and a variety of DES control laws, including those that are quantitatively analyzed using a language measure.

Acknowledgments

The authors would like to acknowledge Dr. Devendra Tolani for his support during the work reported in this paper. This work has been supported in part by the US Army Research Laboratory and the US Army Research Office under Grant No. DAAD19-01-1-0646 and by NASA Glenn Research Center under Grant No. NNC04GA49G.

Appendix Language measure and discrete event optimal control

This appendix reviews the previous work on language measure (Ray, 2005; Ray et al., 2005) and optimal control policy (Ray et al., 2004) that is based on this measure with no event disabling penalty. The background information necessary to develop a performance index for the optimal DES control law is introduced. Performance of DES controllers is associated with the language measure through out this paper.

Let the dynamical behavior of a physical plant be modeled as a deterministic finite state automaton (DFSA) $G_i \equiv (Q, \Sigma, \delta, q_i, Q_m)$, where Q is the finite set of states q_j with $|Q| = n$ and $q_i \in Q$ is the initial state; Σ is the (finite) alphabet of events with $|\Sigma| = m$; the function $\delta : Q \times \Sigma \rightarrow Q$ represents state transitions and $Q_m \subseteq Q$ is the set of marked states which have some importance (positive or negative) for the DFSA model.

Definition 1. The characteristic state weight function that assigns a signed real weight to states is defined as: $\chi : Q \rightarrow [-1,\ 1]$ such that

$$\chi_j \equiv \chi(q_j) \in \begin{cases} [-1,\ 0) & \text{if } q_j \in Q_m^-, \\ \{0\} & \text{if } q_j \notin Q_m, \\ (0,\ 1] & \text{if } q_j \in Q_m^+, \end{cases}$$

where $Q_m^- \subseteq Q$ and $Q_m^+ \subseteq Q$ are the set of negatively and positively marked states respectively. The $(n \times 1)$ characteristic vector is denoted as

$$\bar{\chi} \equiv [\chi_1\ \ \chi_2\ \ \cdots\ \ \chi_n]^{\mathrm{T}}.$$

Definition 2. The event cost is the relative frequency of occurrence of an event given the DFSA state at which the event is generated, and is defined as

- $\pi[\sigma_k |\, q_j] \equiv \tilde{\pi}_{jk} \in [0,\ 1)$, relative frequency of occurrence of event k at state j;
- $\tilde{\pi}[\sigma_k|q_j] = 0$ if $\delta(q_j, \sigma_k)$ is undefined, i.e., event k is not defined at state j.

The $(n \times m)$ event cost matrix is denoted as $\tilde{\Pi} \equiv [\tilde{\pi}_{ij}]$.

Definition 3. The state transition cost of the DFSA is a function $\pi : Q \times Q \rightarrow [0, 1)$ defined as the relative frequency of transition from state j to state k such that $\pi(q_k|q_j) = \sum \tilde{\pi}(\sigma|q_j) \equiv \pi_{jk}$ and the $n \times n$ state transition cost matrix, denoted as Π-matrix, is defined as

$$\Pi = \begin{bmatrix} \pi_{11} & \pi_{12} & \cdots & \pi_{1n} \\ \pi_{21} & \pi_{22} & \cdots & \pi_{2n} \\ \vdots & & \ddots & \vdots \\ \pi_{n1} & \pi_{n2} & \cdots & \pi_{nn} \end{bmatrix}.$$

Note that event costs and state transition costs are very much similar to probabilities and probabilistic interpretation is given in Ray et al. (2005). In this sense, an event cost can be analogous to the probability of an event to occur at a state, and a state transition cost can be analogous to the probability of leaping from one state to another.

Now we define the language measure in terms of the signed state weight function χ and the non-negative state transition cost π.

Definition 4. The signed real measure of the language $L(G_i)$ created by a DFSA G_i, initialized at the state $q_i \in Q$, is defined as

$$\mu_i \equiv \mu(L(G_i)).$$

The $(n \times 1)$ real signed measure vector is denoted as

$$\bar{\mu} \equiv [\mu_1\ \ \mu_2\ \ \cdots\ \ \mu_n]^{\mathrm{T}}.$$

It has been shown in Ray (2005), and Ray et al. (2005) that the measure of the language $L(G_i)$, where $G_i = (Q, \Sigma, \delta, q_i, Q_m)$ can be expressed as $\mu_i = \sum_j \pi_{ij}\, \mu_j + \chi_i$. Equivalently, in vector notation: $\bar{\mu} = \Pi \bar{\mu} + \bar{\chi}$. Therefore, the measure vector $\bar{\mu}$ is uniquely determined as

$$\bar{\mu} = [I - \Pi]^{-1} \bar{\chi}.$$

In the gas turbine engine application, the penalty of disabling controllable events (e.g., redistribution of thrust between two engines, and nozzle area reduction for individual engines) is set to zero as these manipulations do not require any special effort.

The state-based optimal control policy is obtained by selectively disabling controllable events to maximize the measure of the controlled plant language. In each iteration, the optimal control algorithm attempts to disable all controllable events leading to "negatively marked states" and enable all controllable events leading to "positively marked states". It has been also shown in Ray et al. (2004, 2005) that computational complexity of the control synthesis is polynomial in the number of plant states.

The algorithm for synthesis of the optimal control policy is summarized as follows: let G be the DFSA plant model without any constraint of operational specifications. Let the state transition cost matrix of the unsupervised plant be: $\Pi^{plant} \in \Re^{n \times n}$ and the characteristic vector be: $\bar{\chi} \equiv [\chi_1\ \ \chi_2\ \ \cdots\ \ \chi_n]^{\mathrm{T}}$. Starting with iteration index $k = 0$, and $\Pi^0 \equiv \Pi^{plant}$, the control policy is constructed by the following two-step procedure:

Step 1: For every state q_j for which $\mu_j^0 < 0$, disable controllable events leading to q_j. Now, $\Pi^1 = \Pi^0 - \Delta^0$, where $\Delta^0 \geq 0$ is composed of event costs corresponding to all controllable events that have been disabled at $k = 0$.

Step 2: Starting with $k = 1$, if $\mu_j^k \geq 0$, re-enable all controllable events leading to q_j, which were disabled in Step 1. The cost matrix is updated as: $\Pi^{k+1} = \Pi^k + \Delta^k$ for $k \geq 1$, where $\Delta^k \geq 0$ is composed of event costs corresponding to all currently re-enabled controllable events. The iteration is terminated when no controllable event leading to q_j remains disabled for which $\mu_j^k \geq 0$, i.e., if there is no sign change in the measure vector $\bar{\mu}$ between two consecutive iteration steps. At this stage, the optimal value of the performance index, which is the language measure, is $\bar{\mu}^* = [I - \Pi^*]^{-1} \bar{\chi}$.

References

Basseville, M., & Nikiforov, I. V. (1993). *Detection of abrupt changes: Theory and application*. New Jersey: PTR Prentice-Hall.

Cassandras, C. G., & Lafortune, S. (1999). *Introduction to discrete event systems*. Norwell, MA: Kluwer Academic.

DeLaat, J. C., & Chang, C. T. (2003). Active control of high frequency combustion instability in aircraft gas-turbine engines. *International symposium on air breathing engines*, Cleveland, OH, 2003.

Diao, Y., & Passino, K. M. (2001). Stable fault-tolerant adaptive fuzzy/neural control for a turbine engine. *IEEE Transactions on Control Systems Technology*, *9*(3), 494–509.

Fu, J., Yasar, M., & Ray, A. (2004). Optimal discrete event supervisory control of aircraft gas turbine engines. *Proceedings of American Control Conference*, *6*, 5710–5715.

Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2001). *Introduction to automata theory*, *languages*, *and computation* (2nd ed). Reading, MA: Addison-Wesley.

Kumar, R., & Garg, V. K. (1995). *Modeling and control of logical discrete event systems*. Norwell, MA: Kluwer Academic ISBN 0-7923-9538-7.

Litt, J. S., Parker, K. I., & Chatterjee, S. (2003). Adaptive gas turbine engine control for deterioration compensation due to aging. *International symposium on air breathing engines*, Cleveland, OH, 2003.

Ljung, L. (1999). *System identification*: *Theory for the user* (2nd ed). New Jersey: Prentice-Hall.

Parker, K. I., & Guo, T. H. (2002). Development of a turbofan engine simulation in a graphical simulation environment. *JANNAF 26th aeropropulsion subcommittee meeting*, Destin, FL, 2002.

Ramadge, P. J., & Wonham, W. M. (1987). Supervisory control of a class of discrete event processes. *SIAM Journal on Control and Optimization*, *25*(1), 206–230.

Ray, A. (2005). Signed real measure of regular languages for discrete event supervisory control. *International Journal of Control*, *78*(12), 949–967.

Ray, A., Fu, J., & Lagoa, C. (2004). Optimal supervisory control of finite state automata. *International Journal of Control*, *77*(12), 1083–1100.

Ray, A., Phoha, V. V., & Phoha, S. (2005). *Quantitative measure for discrete event supervisory control*. New York: Springer ISBN 0-387-02108-6.

Tolani, D. K., Yasar, M., Ray, A., & Yang, V. (2006). Anomaly detection in aircraft gas turbine engines. *Journal of Aerospace Computing, Information, and Communication*, *3*(2), 44–51.

Volponi, A. J., Brotherton, T., Luppold, R., & Simon, D. L. (2003). Development of an information fusion system for engine diagnostics and health management. *JANNAF 27th airbreathing propulsion subcommittee meeting*, Colorado Springs, CO, 2003.

Yasar, M., Horn, J. F., & Ray, A. (2006). Effects of supervisory decisions on nonlinear aircraft dynamics. *Proceedings of American control conference*, 4154–4159.